

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jakob Košir

**Modeliranje odnosa med treningi in
zmogljivostjo v vzdržljivostnih športih**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Erik Štrumbelj

Ljubljana, 2016

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Modeliranje odnosa med treningi in zmogljivostjo v vzdržljivostnih športih

Tematika naloge:

Cilj naloge je raziskava metod za napovedovanje zmogljivosti športnikov na podlagi podatkov o treningih in rezultatih na preteklih testiranjih. Primerjali bomo obstoječi metodi Banisterjev model in Performance Potential model, Banisterjev model pa bomo poskušali tudi izboljšati z bayesovskim pristopom k ocenjevanju parametrov. Naloga bo sestavljena iz pregleda in sinteze literature, opisa obstoječih in razvitih metod ter ocenjevanja kvalitete metod na podatkih iz kolesarstva.

Za strokovno podporo, napotke in pomoč pri pridobivanju podatkov se zahvaljujem svojemu mentorju, doc. dr. Eriku Štrumblju, Inštitutu za šport, Miranu Kavašu in Gregorju Sikošku, brez katerih to diplomsko delo ne bi moglo nastati. Zahvaljujem se tudi družini in prijateljem, ki so mi stali ob strani tekom študija in pri izdelavi diplomskega dela.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Cilji	2
2	Podatki	3
2.1	Training Stress Score	4
2.2	Pretvorba podatkov	4
3	Banisterjev model	7
3.1	Omejitve vrednosti parametrov	10
4	Prilagajanje modela	11
4.1	Minimizacija funkcije napake	12
4.2	Implementacija v R	13
5	Vrednotenje modela	17
5.1	Rezultati pri samostojni uporabi metode L-BFGS-B	17
5.2	Primerjava rešitev minimizacije	21
5.3	Preizkus modela na podatkih iz članka	27
5.4	Rezultati	28
6	Sklepne ugotovitve	31
6.1	Primerjava z ostalimi študijami	31
6.2	Možnosti za nadaljnje delo	32

Seznam pojmov

Moč na laktatnem pragu (angl. Functional threshold power, FTP)

Maksimalna moč, ki jo je kolesar sposoben vzdrževati eno uro.

Normalizirana povprečna moč (angl. Normalized Power, NP) Mera

moči, ki zahtevnost kolesarskega treninga opiše bolje kot povprečna moč.

Faktor intenzivnosti (angl. Intensity Factor, IF) Razmerje med nor-

malizirano povprečno močjo in močjo na laktatnem pragu.

Hitrost vzpenjanja (ital. Velocità ascensionale media, VAM) Povprečna

hitrost vzpenjanja v metrih nadmorske višine na uro.

L-BFGS-B Algoritem za optimizacijo Limited-memory Broyden-Fletcher-

Goldfarb-Shanno.

VO₂ max Maksimalna aerobna kapaciteta, največja količina kisika, ki jo

športnik lahko porabi v eni minuti.

TSS Metoda ocenjevanja napora treningov Training Stress Score, temelji na

podatkih o moči.

TRIMP Metoda ocenjevanja napora treningov Training Impulse, temelji na

podatkih o frekvenci srčnega utripa.

Povzetek

Naslov: Modeliranje odnosa med treningi in zmogljivostjo v športu

Na področju športnega treniranja je za izboljšanje posameznikove športne zmogljivosti izredno pomembno predpisovanje učinkovitih načrtov treningov. V želji po boljšem razumevanju spreminjanja športne zmogljivosti so odnos med treningi in zmogljivostjo poskušali opisati z matematičnimi modeli. V tem diplomskem delu je predstavljen Banisterjev model in metoda za ocenjevanje napora treningov "Training Stress Score". Opravljen je tudi pregled obstoječih študij na temo Banisterjevega modela in preizkus kvalitete napovedi modela na podatkih o treningih in testiranjih zmogljivosti šestih cestnih kolesarjev. Kvaliteta napovedi se je izkazala za slabo, med preizkusom so predstavljene tudi slabosti modela.

Ključne besede: športno treniranje, matematično modeliranje, Banisterjev model, matematična optimizacija.

Abstract

Title: Modeling the relationship between physical training and performance in endurance sports

Exercise regimen is an integral part of sports coaching, the goal of which is to improve an individual's athletic performance. In an effort to better understand and optimize athletic performance, exercise physiologists have developed mathematical models of physical training and performance. Banister model and "Training Stress Score", a method to quantify training load have been presented in this thesis. An overview of existing literature has been conducted and Banister model has been tested on training data of six road cyclists. Model's prediction quality has been proven poor and its shortcomings have been exposed during evaluation.

Keywords: sports coaching, mathematical modelling, Banister model, mathematical optimization.

Poglavje 1

Uvod

Sposobnost predpisovanja učinkovitih načrtov treningov je bila zgodovinsko najbolj pogojena z izkušnostjo športnih trenerjev. Sodobnejši pristop je uporaba znanstvenih metod, ki temeljijo na razumevanju odnosa med treningi, fiziološkimi prilagoditvami, ki jih ti povzročajo, in učinka teh prilagoditev na športno zmogljivost posameznika.

Ena od teh metod je modeliranje odnosa med treningi in zmogljivostjo posameznega športnika oziroma napovedovanje zmogljivosti iz podatkov o treningih. Obstajajo matematični modeli, ki športnika obravnavajo kot sistem, ki ima na vhodu podatke o treningih, na izhodu pa različne komponente učinkov treningov. Negativne, kot npr. utrujenost, in pozitivne, fiziološke prilagoditve, ki izboljšujejo športno zmogljivost posameznika. Končno napoved zmogljivosti nato dobimo z interakcijo teh komponent. Vrednosti komponent se s časom spreminjajo (padajo), v modelih se kot najmanjša časovna enota tipično uporablja 1 dan.

Modeliranje učinka treningov na zmogljivost športnikov omogoča pripravo boljših in bolj individualiziranih načrtov treningov. Trenerjem pomaga predvsem pri časovni razporeditvi treningov, za doseganje najboljše forme takrat, ko bi jo želeli. Tempiranje forme velja za enega najtežjih delov sestavljanja načrtov treningov, ki bi ga lahko z uspešnim modeliranjem zmogljivosti enostavno rešili.

V literaturi se najbolj pogosto pojavlja Banisterjev model, ki ga je predlagal dr. Eric Banister leta 1976 [4]. Na temo tega modela obstaja veliko študij, kjer so z njim napovedovali zmogljivost športnikov v več različnih športih. Napovedi so se izkazale za dobre, ampak le če je bilo na voljo dovolj rezultatov testiranja zmogljivosti [6].

1.1 Cilji

Na podatkih treningov cestnih kolesarjev bomo preverili kvaliteto napovedi Banisterjevega modela. Napovedovali bomo rezultat zadnjega testiranja in ga primerjali z izmerjeno vrednostjo. Kot mero zmogljivosti uporabljamo *Moč na laktatnem pragu* (angl. *Functional threshold power, FTP*).

Študije s tega področja se osredotočajo predvsem na uporabnost napovedi modelov in le na kratko opišejo matematičen postopek prilagajanja modela posamezniku. Ta postopek bomo bolj natančno opisali in napoved modelov skušali izboljšati z uporabo sodobnejših statističnih metod.

Prvotno smo želeli analizirati tudi model Performace Potential, zaradi omejene količine vsebine pa smo se osredotočili le na Banisterjev model.

Poglavje 2

Podatki

Za preizkus modelov potrebujemo čim več podatkov o treningih in testiranjih zmogljivosti športnikov v nekem časovnem obdobju. Ker se lastnosti športnika in s tem vrednosti parametrov modela čez čas spreminjajo [8], modele učimo in testiramo samo na podatkih ene sezone. V ostalih raziskavah so uspeli z Banisterjevim modelom napovedovati zmogljivost v različnih športih in z uporabo različnih načinov ocenjevanja naporov treningov. V nalogi se bomo omejili le na kolesarstvo, predvsem zaradi dostopnosti podatkov.

Za uporabo v matematičnih modelih moramo treninge številsko ovrednotiti. Vsakemu treningu želimo prirediti neko številsko oceno, v kateri skušamo zajeti lastnosti, ki vplivajo na fiziološke prilagoditve športnika in s tem na spremembo zmogljivosti. Obstaja več metod za ocenjevanje napora treningov, v kolesarstvu pa se največ uporabljajo take, ki temeljijo na frekvenci srčnega utripa ali moči. V podatkih, ki jih uporabljamo v nalogi, so treningi ocenjeni z metodo *Training Stress Score* (TSS), ki jo opišemo v podpoglavju 2.1.

Za nalogo smo uspeli pridobiti podatke o treningih in testiranjih šestih kolesarjev, za vsakega po eno sezono. Gre za slovenske kolesarje, od boljših amaterjev do poklicnih kolesarjev, ki tekmujejo v različnih kontinentalnih ali profesionalnih ekipah. Za vsakega imamo na voljo datume treningov in njihove vrednosti TSS ter rezultate od 4 do 15 testiranj FTP.

2.1 Training Stress Score

Ocena temelji na *Normalizirani povprečni moči* (angl. *Normalized Power*, *NP*) in *Faktorju intenzivnosti* (angl. *Intensity Factor*, *IF*). Zaradi visoke variabilnosti moči med kolesarjenjem sama povprečna moč ni dober indikator zahtevnosti treninga. Koncept normalizirane moči temelji na naslednjih dejstvih:

- Fiziološki odzivi na spremembe intenzivnosti niso takojšnji, kljub temu pa so predvidljivi.
- Mnogi pomembni fiziološki odzivi niso linearno odvisni od intenzivnosti.

Normalizirana povprečna moč še vedno ni dovolj za oceno napora treninga, saj ne upošteva fizične pripravljenosti posameznika. Za to uporabimo *IF*, ki ga izračunamo kot razmerje med *NP* in *FTP*.

Pri izračunu *TSS* upoštevamo še trajanje treninga, kjer je *s* trajanje treninga v sekundah [8]:

$$\text{TSS} = 100 \cdot \frac{s \cdot \text{NP} \cdot \text{IF}}{3600 \cdot \text{FTP}} \quad (2.1)$$

2.2 Pretvorba podatkov

Za lažjo obdelavo smo vse podatke pretvorili v obliko CSV z naslednjimi stolpci:

id_kolesarja	datum	TSS	FTP
--------------	-------	-----	-----

Pri štirih kolesarjih so podatki v obliki delovnega zvezka Excel. Ročno smo izbrali relevantne stolpce (datum, TSS in FTP) in jih izvozili v obliko CSV.

2.2.1 Izračun vrednosti FTP za kolesarja brez merilca moči

Eden od kolesarjev ni imel merilca moči, testiranja je zato opravljal z vožnjo v klanec. Pri dovolj strmih klancih zunanji dejavniki (kot npr. veter) nimajo veliko vpliva na hitrost, zato lahko dovolj natančno izračunamo povprečno moč kolesarja. Moč izračunamo iz podatka o povprečnem naklonu klanca, vrednosti *Hitrosti vzpenjanja* (ital. *Velocità ascensionale media*, VAM) in mase kolesarja. Testiranja je opravljal na klanecu *Zelena magistrala*, s povprečenim naklonom 8% [15].

Najprej izračunamo relativno moč kolesarja v $[\frac{W}{kg}]$:

$$\text{Relativna moč} = \frac{\text{VAM}}{100 \cdot (2 + \frac{\text{naklon}}{10})} \quad (2.2)$$

Absolutno moč v [W] dobimo tako da relativno moč $[\frac{W}{kg}]$ pomnožimo z maso kolesarja. Podatka o masi kolesarja nimamo, predpostavimo, da ima 75 kg. Tako dobimo oceno povprečne moči te vožnje, iz katere lahko izračunamo oceno FTP. Kolesar je opravljal testiranje z intenzivnostjo *Tempo*, približno 90% FTP moči. Moč iz testiranja tako delimo z 0.9, da dobimo oceno moči FTP [9].

2.2.2 Izračun vrednosti TSS in FTP iz datotek kolesarskih računalnikov

Za enega od kolesarjev imamo na voljo vse datoteke izvožene iz kolesarskih računalnikov. Te datoteke vsebujejo podatke o geografski lokaciji, hitrosti, frekvenci srčnega utripa, moči in kadenci (število obratov pedal na minuto) celotnih treningov. Podatki so zapisani v sekundnih intervalih.

Zanima nas le vrednost TSS vsakega treninga in časovni intervali v treningih, iz katerih lahko ocenimo trenutno vrednost FTP. Take intervale in vrednosti TSS najlažje poiščemo s programom za analizo kolesarskih treningov WKO [17]. Iščemo 5 in 20-minutne intervale, ki jih je kolesar prevozil

z maksimalno možno močjo. Oceno vrednosti FTP iz 5-minutnega intervala dobimo tako, da povprečno moč pomnožimo z 0.8, povprečno moč 20-minutnih intervalov pa množimo z 0.95 [9].

Poglavje 3

Banisterjev model

V literaturi se na področju modeliranja odnosa med treningi in zmogljivostjo največkrat pojavlja Banisterjev "impulse-reponse model", ki ga je predlagal Dr. Eric Banister leta 1976 [4]. V tem članku sprva predlagajo modeliranje zmogljivosti na podlagi štirih njenih sestavnih komponent:

- vzdržljivost,
- moč,
- spretnost,
- fiziološki dejavniki.

Vpliv posameznih komponent na dosežene rezultate je odvisen predvsem od športa, pri dviganju uteži je najbolj pomembna moč, pri teku pa vzdržljivost. Dodatno imamo še komponento utrujenosti, ki na zmogljivost vpliva negativno. Interakcijo med komponentami za napoved zmogljivosti je težko določiti, njihove vrednosti bi lahko seštevali, množili... Zato predlagajo več različic modela, tudi takega z eno samo komponento zmogljivosti, ki predstavlja vse pozitivne učinke treninga in model z dvema komponentama; ta ima poleg pozitivne še komponento utrujenosti. Ta se je izkazal za najbolj natančnega [2] in tudi v drugih študijah uporabljajo različico z dvema komponentama.

V modelu so učinki treninga modelirani s prenosno funkcijo med vhodi in izhodi. Na vhodu ima funkcija dnevne doze treninga, izražene npr. s TSS, na izhodu pa vrednosti komponent. Vsak trening ima nek pozitiven učinek na zmogljivost, povzroča pa tudi utrujenost, ki negativno vpliva na zmogljivost. Iz interakcije med komponentama določimo napoved zmogljivosti, npr. FTP ob nekem času. V tej različici modela za napoved preprosto odštejemo vrednost komponente utrujenosti od pozitivne.

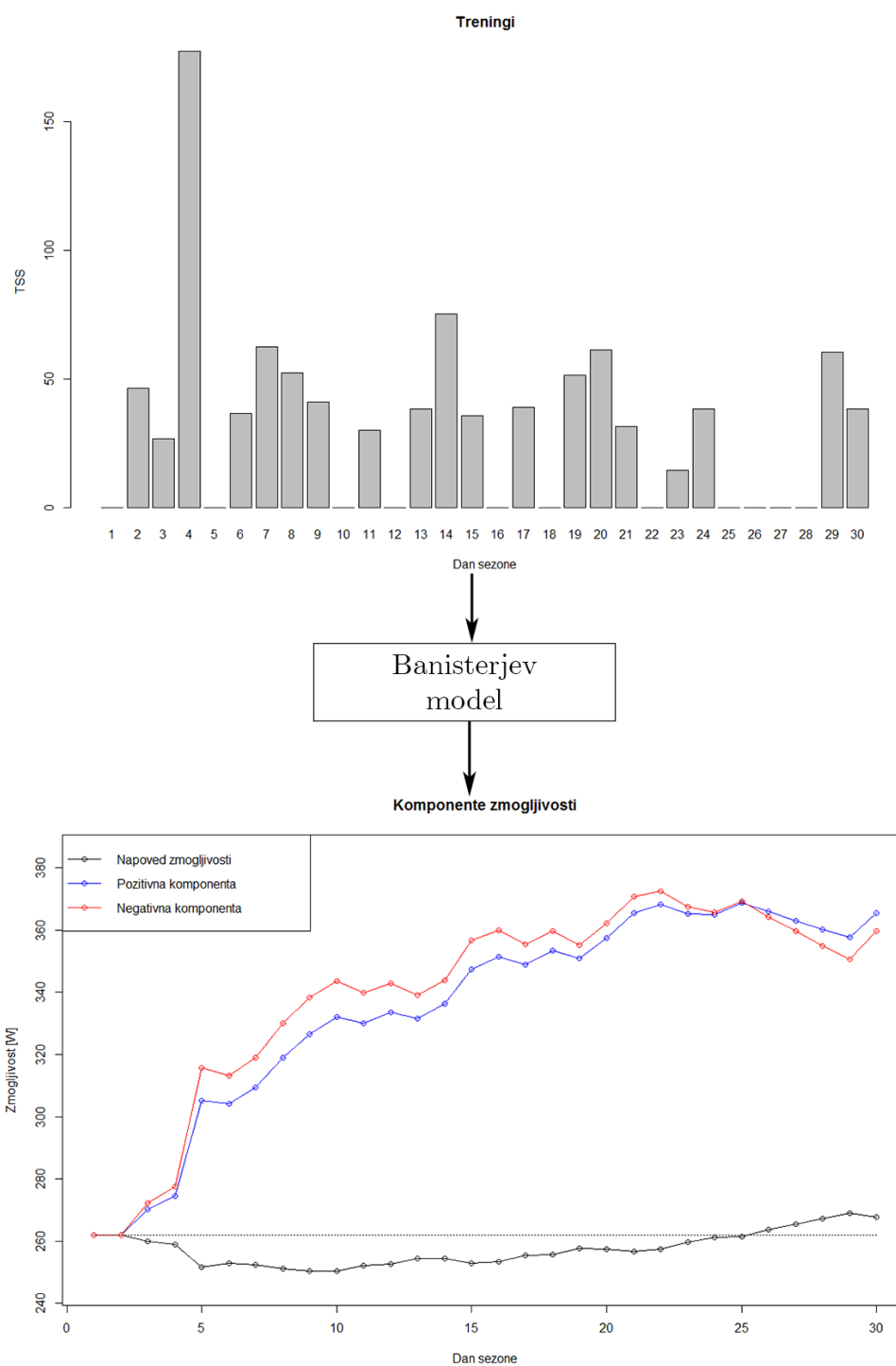
Učinek obeh komponent v času eksponentno pada (z različnima eksponentoma), velikost obeh komponent pa model uravnava s še dvema parametroma:

$$p_t = p_0 + k_a \sum_{s=0}^{t-1} w_s e^{-(t-s)/\tau_a} - k_f \sum_{s=0}^{t-1} w_s e^{-(t-s)/\tau_f}, \quad (3.1)$$

p_t je napoved zmogljivosti posameznika na dan t , p_0 začetna zmogljivost (na dan $t = 0$), k_a in k_f sta koeficienta velikosti pozitivne in negativne komponente, τ_a in τ_f časovni konstanti, ki določata hitrost padanja pozitivne in negativne komponente skozi čas, w_s pa količina treninga na dan s .

Parametra k_a in k_f poleg uravnavanja velikosti učinka obeh komponent služita še prilagajanju modela na različne načine ocenjevanja napora treningov. Vrednosti parametrov k_a in k_f pri modelu istega kolesarja bi bile različne, če bi namesto TSS uporabili npr. *Training Impulse* (TRIMP) oceno.

Model je bil uspešno uporabljen v različnih športih, vključujoč dvigovanje uteži, met kladiva, tek, plavanje, kolesarstvo, triatlon... Na podlagi podatkov o treningih lahko z njim pojasnimo vsaj 70%, pogosto več kot 90% sprememb v zmogljivosti posameznika [8].



Slika 3.1: Shema Banisterjevega modela. Z modelom napovemo pozitivno in negativno komponento, ki jih odštejemo za končno napoved zmogljivosti. Ko je pozitivna komponenta višja od negativne, je zmogljivost višja od začetne.

3.1 Omejitve vrednosti parametrov

Če za parametre modela velja,

- da je k_a manjši od k_f
- in τ_f manjši od τ_a ,

je kratkoročni učinek na športno zmogljivost posameznika negativen (zaradi utrujenosti), dolgoročni pa pozitiven, saj velikost pozitivne komponente pada počasneje.

Nekateri zgornja pogoja obravnavajo kot obvezna [8], drugi pa jih ne omenjajo in ne upoštevajo. Če bi lahko natančno modelirali utrujenost in vplive dejavnikov, ki izboljšujejo zmogljivost, se zdi logično, da za vsak model veljata. V to dvomimo, zato se v nalogi ne bomo omejili samo na modele, ki pogojemata ustrezajo. Zanima nas, če lahko tudi tako dosežemo kvalitetno napoved.

Poglavje 4

Prilagajanje modela

Štiri parametre modela želimo za vsakega kolesarja izbrati tako, da bo z modelom mogoče kar najbolje napovedovati zmogljivost tega kolesarja. Napoved zmogljivosti naj bo na dan testiranja kar najbolj podobna dejanskemu rezultatu tega testiranja.

Podatke vsakega kolesarja razdelimo na 2 množici, učno in testno. Učna množica vsebuje podatke o vseh testiranjih FTP razen zadnjega in treningih do predzadnjega testiranja. Na tej množici bomo določili parametre modela. Kvaliteto modela ocenimo tako, da na podlagi podatkov o vseh treningih napovemo zmogljivost na dan zadnjega testiranja in rezultat primerjamo s pravo vrednostjo.

Med učenjem kvaliteto napovedi ocenimo glede na napako (razliko med napovedjo in dejanskim rezultatom) vseh testiranj. Napake želimo sešteti, da pa se izognemo negativnim vrednostim, jih najprej še kvadriramo (s tem tudi bolj kaznujemo večje napake). Imamo funkcijo napake, katere vrednost je vsota kvadratov napak za podane parametre modela. Najboljšo napoved poiščemo tako, da najdemo množico parametrov, za katero je vrednost funkcije napake najmanjša.

4.1 Minimizacija funkcije napake

Za iskanje parametrov, pri katerih je vrednost funkcije napake najmanjša, uporabimo metodo *Limited-memory Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS-B) [3].

L-BFGS-B je kvazi-Newtonova metoda za reševanje nelinearnih optimizacijskih problemov. Take probleme matematično opišemo kot

$$x^* = \arg \min_x f(x), \quad (4.1)$$

kjer je $f : \mathbb{R}^n \rightarrow \mathbb{R}$ parcialno odvedljiva nelinearna funkcija.

V primeru prilagajanja Banisterjevega modela minimiziramo funkcijo napake:

$$S = \sum_{i=0}^n (y_i - \text{Banister}(x_i, k_a, k_f, \tau_a, \tau_f))^2, \quad (4.2)$$

kjer je y_i rezultat testiranja na dan x_i , $\text{Banister}(x_i, k_a, k_f, \tau_a, \tau_f)$ pa napoved modela na dan x_i .

Metoda za iskanje najnižje točke uporablja gradient. Gradient je vektor, ki kaže v smeri najhitrejšega naraščanja funkcije, v minimumih in maksimumih funkcije pa je enak nič. Pri računanju lahko uporabljamo numerični približek gradienta, metoda pa rešitev najde hitreje, če gradient izračunamo analitično.

Gradient funkcije izračunamo kot vektor parcialnih odvodov po vseh spremenljivkah:

$$\nabla f = \left(\frac{df}{dx_1}, \frac{df}{dx_2}, \dots, \frac{df}{dx_n} \right). \quad (4.3)$$

Gradient funkcije napake je torej:

$$\nabla S = \sum_{t=1}^n \sum_{s=0}^{t-1} (y_t - \text{Banister}(x_t, k_a, k_f, \tau_a, \tau_f)) \begin{bmatrix} -2w_s e^{-\frac{(t-s)}{\tau_a}} \\ 2w_s e^{-\frac{(t-s)}{\tau_f}} \\ -\frac{2e^{-\frac{(t-s)}{\tau_a}} k_a (t-s) w_s}{t_a^2} \\ \frac{2e^{-\frac{(t-s)}{\tau_f}} k_f (t-s) w_s}{t_f^2} \end{bmatrix}^T. \quad (4.4)$$

4.1.1 Upoštevanje omejitev parametrov

Če pri minimizaciji želimo upoštevati predlagane omejitve Banisterjevega modela (iz poglavja 3.1) lahko to naredimo tudi z metodo L-BFGS-B tako, da model re-parametriziramo.

Metoda L-BFGS-B omogoča omejitve vrednosti parametrov z zgornjo in spodnjo mejo v obliki $a_i \leq x_i \leq b_i$, kjer sta a_i in b_i poljubni številski konstanti, x_i pa parameter funkcije, ki jo minimiziramo.

Da uveljavimo pogoj $k_a \leq k_f$ v enačbi modela parameter k_f zamenjamo z $k_a + \Delta k$, podobno zamenjamo τ_a z $\tau_f + \Delta\tau$. Dobimo re-parametrizirano enačbo modela:

$$p_t = p_0 + k_a \sum_{s=0}^{t-1} w_s e^{-(t-s)/(\tau_f + \Delta\tau)} - (k_a + \Delta k) \sum_{s=0}^{t-1} w_s e^{-(t-s)/\tau_f}, \quad (4.5)$$

s parametri $\{k_a, \Delta k, \tau_f, \Delta\tau\}$. Nato optimizacijski metodi postavimo pogoja, da sta Δk in $\Delta\tau$ pozitivni števili. Po dobljeni rešitvi optimizacije parametre zamenjamo nazaj, $k_f = k_a + \Delta k$ in $\tau_a = \tau_f + \Delta\tau$, da lahko uporabljamo izvorno enačbo modela (3.1).

4.2 Implementacija v R

R je programski jezik in okolje prilagojeno statističnemu delu [14]. Vsebuje implementacije pogosto uporabljenih statističnih in drugih računskih algoritmov, med drugim tudi več metod za minimizacijo funkcij.

Za minimizacijo funkcije napak najprej implementiramo funkcijo, ki z izračunom enačbe 3.1 ob podanih parametrih modela vrne napoved zmogljivosti na dan t .

V R je računanje z vektorji veliko hitrejšo kot operacije nad posameznimi elementi vektorjev. Pri izračunu vsot (npr. $\sum_{s=0}^{t-1} w_s e^{-(t-s)/\tau_a}$) zato najprej izračunamo vektor $e^{-(t-s)/\tau_a}$, $s = 0 \dots t-1$, ga množimo z vektorjem vrednosti TSS in z funkcijo `sum()` seštejemo elemente.

```
banister <- function(df, t, ka, kf, ta, tf) {  
  # Banisterjev model, vrne napoved zmogljivosti na dan t  
  # df je tabela podatkov o kolesarju s stolpci:  
  # datum, TSS, FTP  
  
  # Začetni FTP, rezultat prvega testiranja  
  p0 = df$FTP[1]  
  
  # Vektor od 1 do t brez t  
  # napoved na dan t upošteva treninge do t-1,  
  # na dan 1 je napoved enaka p0  
  s = head(1:t,-1)  
  
  # Izračunamo pozitivno in negativno komponento  
  pozitivna = sum(exp(-(t - s) / ta) * df[s, 'TSS'])  
  negativna = sum(exp(-(t - s) / tf) * df[s, 'TSS'])  
  
  p0 + ka * pozitivna - kf * negativna  
}
```

Nato implementiramo funkcijo napak, ki jo bomo optimizirali:

```
sum_square_errors <- function(params, df) {  
  # Funkcija napak, vsota kvadratov napak v točkah testiranja  
  # df je tabela podatkov o kolesarju s stolpci:  
  # datum, TSS, FTP  
  
  ka = params[1]  
  kf = params[2]  
  ta = params[3]  
  tf = params[4]  
  
  # Napovedi modela v točkah testiranja  
  napovedi = banister_v(df, which(meritve_sel), ka, kf, ta, tf)  
  prave_vrednosti = df[meritve_sel, 'FTP']  
  # Vsota kvadratov napak v točkah testiranja  
  sum((napovedi - prave_vrednosti) ^ 2)  
}
```

Pred klicem optimizacijske metode za hitrejše izvajanje implementiramo še gradient funkcije napak (`sum_square_errors.grad`) in izberemo začetni približek rešitve (`initial_params`).

```
result = optim(  
  par = initial_params,  
  fn = sum_square_errors,  
  gr = sum_square_errors.grad,  
  method = 'L-BFGS-B',  
  df=df)
```


Poglavje 5

Vrednotenje modela

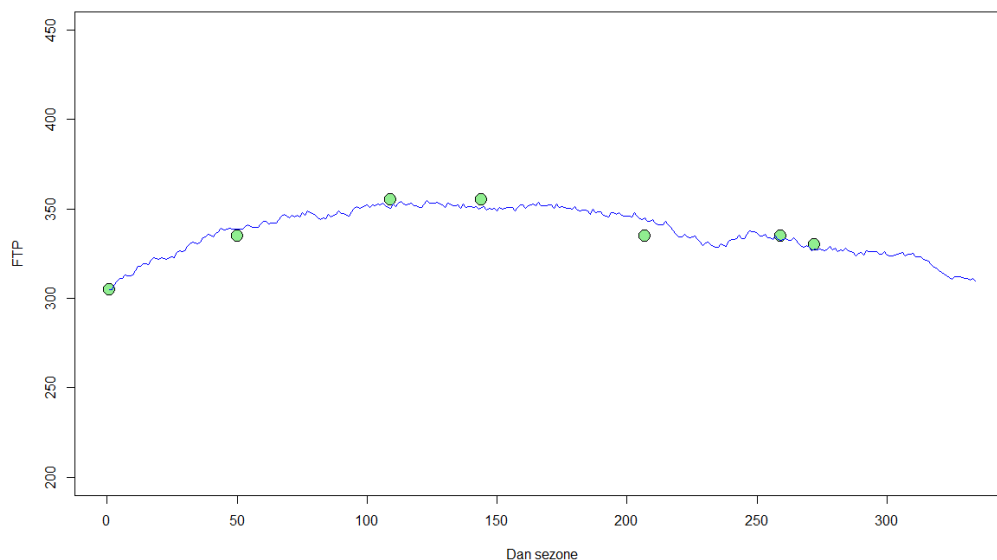
5.1 Rezultati pri samostojni uporabi metode L-BFGS-B

Pri samostojni uporabi metode L-BFGS-B je napoved, ki jo dobimo kot rešitev (minimum) funkcije napake, vsaj na pogled smiselna in se dobro prilega dejanskim rezultatom meritev zmogljivosti (glejte sliko 5.1).

V tem primeru model učimo na vseh meritvah, saj preizkušamo samo pravilnost naše implementacije Banisterjevega modela. Za testiranje kvalitete napovedi bomo kasneje model učili na meritvah v učni množici (vse meritve razen zadnje), nato pa napovedovali rezultat zadnjega testiranja.

Opazimo, da je rešitev minimizacijskega problema močno odvisna od izbire začetnega približka, tj. vrednosti parametrov k_a, k_f, τ_a, τ_f , pri katerih metoda L-BFGS-B začne iskanje. Pri različnih začetnih približkih so rešitve (in s tem napovedi modela) zelo različne. Če začetne parametre slabo izberemo, lahko dobimo tudi zelo slabo napoved, ki močno odstopa od dejanskih rezultatov testiranja, zmogljivost športnika pa se spreminja hitreje, kot je to mogoče. Pri uporabi modela seveda ne želimo, da je za kvalitetno napoved potrebno dobro izbrati začetne parametre.

Zanima nas, koliko različnih rešitev (in napovedi) dobimo z izbiro različnih začetnih parametrov. Optimizacijsko metodo izvedemo z različnimi začetnimi



Slika 5.1: Napovedana zmogljivost (modra črta) in testiranja zmogljivosti (zelene pike) za enega izmed kolesarjev.

parametri, nato pa primerjamo dobljene napovedi. Za različne začetne parametre generiramo kartezični produkt A^4 množice $A = \{5, 15, 25, 35, 45, 55, 65, 75\}$, ki zajema celoten nabor vrednosti parametrov, ki se pojavljajo v literaturi. Tako dobimo $8^4 = 4096$ različnih vektorjev začetnih vrednosti, s katerimi preizkusimo optimizacijsko metodo.

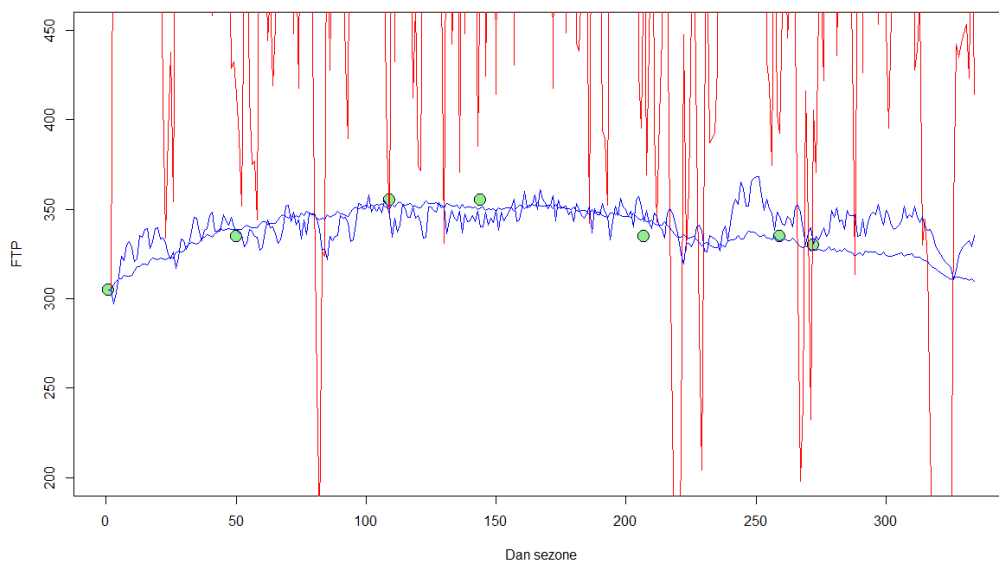
Pri tem si zapomnimo vse dobljene rešitve, ki so si med sabo dovolj različne. Kot dovolj različne upoštevamo tiste, ki se od vseh ostalih po vsotah kvadratov razlik v vseh točkah napovedi razlikujejo za vsaj 10000. Povprečna razlika med dvema različnima napovedma v eni točki mora tako biti $\sqrt{\frac{10.0000}{334}} = 5.47$ (334 je število dni, za katere napovedujemo zmogljivost kolesarja).

Pri vseh kolesarjih dobimo nekaj deset rešitev, odvisno od števila testiranj. Nekaj jih je smiselnih, večina pa je očitno napačnih, saj se zmogljivost spreminja veliko hitreje, kot je to sploh mogoče. Take rešitve lahko dobimo zaradi problema prevelikega prilagajanja ali problema lokalnih minimumov.

5.1. REZULTATI PRI SAMOSTOJNI UPORABI METODE L-BFGS-B19

Problem prevelikega prilagajanja se pogosto pojavlja, ko imamo malo znanih vrednosti ciljne spremenljivke. Model v takem primeru namesto pravega razmerja med podatki opiše napake ali šum. Značilen je dober rezultat (nizka vrednost funkcije napake) na učni množici, na testni množici pa je rezultat slab.

Pri optimizaciji nekonveksnih funkcij je pogost problem lokalnih minimumov. V takih primerih optimizacijska metoda ne najde najboljše rešitve, ampak samo najboljšo lokalno rešitev, tj. najboljša rešitev na nekem omejenem območju preiskovanja. Pri napovedih takih modelov je značilen slab rezultat tako na učni kot tudi na testni množici.



Slika 5.2: Smiselne rešitve (modre črte), očitno napačne rešitve (rdeča črta) in testiranja zmogljivosti (zeleno pike) za enega izmed kolesarjev.

	k_a	k_f	τ_a	τ_f	Vsota kvadratov napak
1	4.99	4.97	82.93	83.20	163
2	0.77	0.74	35.12	36.24	385
3	55.42	55.60	2.77	2.75	468
4	0.09	63.93	6.71	0.15	576
5	43.86	43.91	5.81	5.80	582
6	6.09	6.02	14.66	14.76	617
7	25.03	24.97	10.00	10.00	649
8	0.02	19.83	22.02	0.16	683
9	4.06	4.56	7.12	6.30	3635
10	19.05	16.92	7.53	8.27	79887
			⋮		

Tabela 5.1: Najboljše rešitve optimizacijske metode za enega izmed kolesarjev. urejene naraščajoče po vsoti kvadratov napak. Prvih osem rešitev je smiselnih, ostale so očitno napačne.

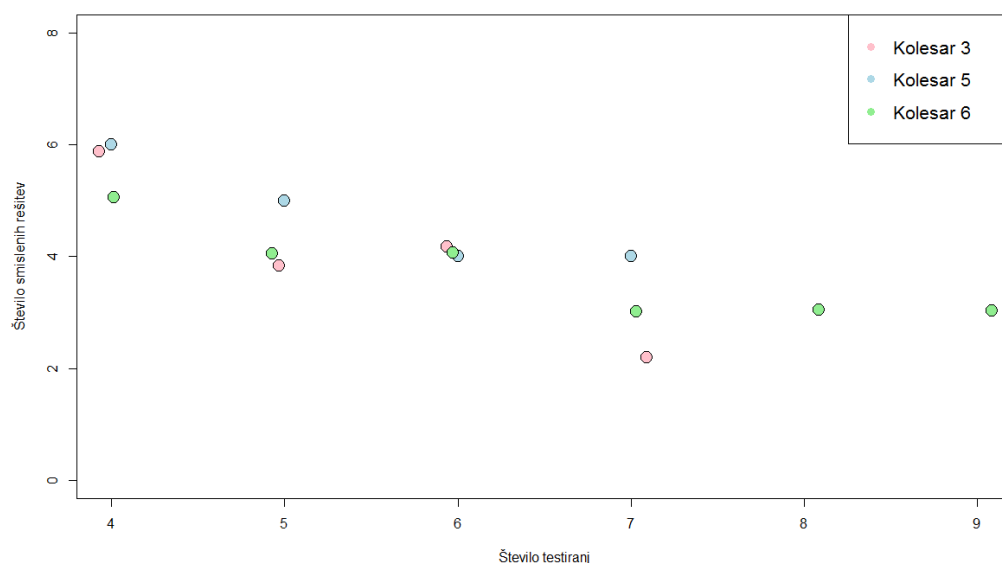
Za enega izmed kolesarjev dobimo 31 različnih rešitev (glej tabelo 5.1), od tega jih je 8 smiselnih, vse ostale pa so očitno napačne. Če rešitve uredimo po vsoti kvadratov napak, opazimo, da je ta pri napačnih rešitvah zelo visoka, torej gre pri teh za problem lokalnih minimumov. To lahko rešimo z boljšo izbiro začetnih parametrov.

Problematično je tudi, da dobimo več smiselnih rešitev, ki imajo podobne vsote kvadratov napak. Vedno želimo najti tisto rešitev (vrednosti štirih parametrov modela), s katerimi bomo na testni množici dosegli najboljše rezultate. To ni nujno tista, ki na učni množici doseže najboljše rezultate. Zato, namesto da izberemo rešitev z najmanjšo vsoto napak, analiziramo lastnosti tistih, ki na testni množici dosegaajo dobre rezultate. Nato lahko že med postopkom optimizacije damo prednost takim rešitvam, npr. z uvedbo regularizacije.

5.2 Primerjava rešitev minimizacije

Ker pri izvajanju optimizacijske metode z izbiro različnih začetnih približkov dobimo več različnih rešitev, želimo najti neke skupne lastnosti tistih, s katerimi lahko dobro napovemo rezultate prihodnjih testiranj zmogljivosti. Za začetek nas zanima, koliko smiselnih rešitev sploh dobimo, če imamo različno število testiranj.

Predvidevamo, da je število rešitev odvisno tudi od samih podatkov in ne samo od števila testiranj, zato primerjamo število rešitev na podatkih več kolesarjev, izberemo 3 z največ testiranjmi FTP, ti jih imajo od 7 do 9. Poiščemo število rešitev pri vseh testiranjih, nato pa odstranimo najbolj “sredinsko” testiranje (ta ima najmanjšo povprečno oddaljenost od vseh ostalih), ponavljamo, dokler ne ostanejo samo 4. Rezultati so vidni na sliki 5.3.



Slika 5.3: Število rešitev glede na število testiranj za tri kolesarje z največ testiranjmi.

Smiselne rešitve definiramo kot rešitve, pri katerih je povprečna razlika med napovedmi in rezultati testiranja manjša od 5 W. Takšna natančnost

je potrebna za praktično uporabnost modela, napoved bi se morala od dejanskega rezultata testiranja razlikovati za največ 2% [8]. Zmogljivost takih rešitev se tudi ne sme spreminjati prehitro (problem prevelikega prilagajanja), kar določimo ročno. Za vse različne rešitve pa mora veljati, da se v povprečju v vsaki točki napovedi razlikujejo za vsaj 5.47 W.

Opazimo, da je število rešitev pri istem številu testiranj lahko različno, vseeno pa je viden trend upadanja z višanjem števila testiranj. Zaradi majhne količine podatkov težko sklepamo na kakršnokoli eksplicitno odvisnost med številom testiranj in številom različnih rešitev, da bi lahko določili, pri koliko testiranjih dobimo samo eno rešitev. Precej nespodbudno izgleda tudi to, da tudi pri velikem številu testiranj (7 do 9) dobimo več različnih rešitev.

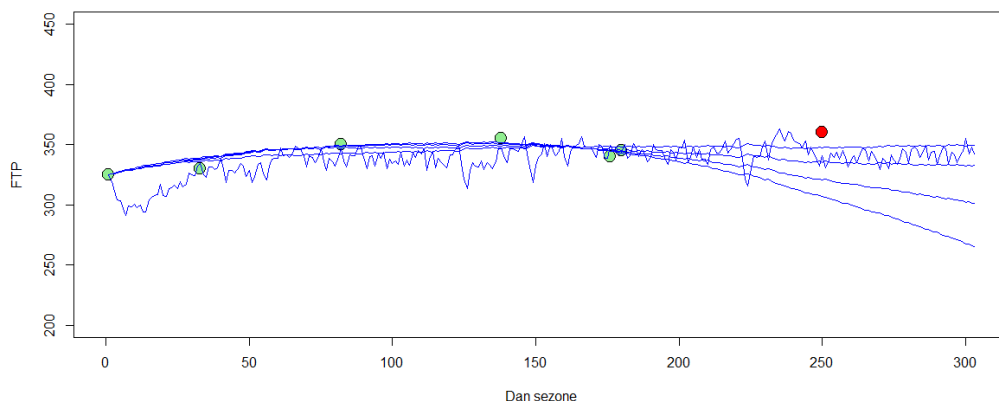
5.2.1 Pregled rešitev po kolesarjih

Za boljši pregled nad napovedmi si ogledamo vse rešitve posameznih kolesarjev. Pri tej primerjavi podatke razdelimo na učno in testno množico, saj želimo primerjati kakovost napovedi na testni množici (zadnje testiranje). Modela seveda ne smemo učiti na rezultatih testiranj, ki jih bomo napovedovali (tj. testna množica).

Pri analizi izpustimo podatke kolesarjev 1 in 4, saj imata samo rezultate štirih testiranj, ostali pa jih imajo med 7 in 9.

Rešitve pri kolesarju 3

V sliki 5.4 vidimo 5 različnih smiselnih rešitev za kolesarja 3. Štiri napovedi, pri katerih se zmogljivost spreminja zelo počasi (ali sploh ne), gotovo ne opišejo pravilno spreminjanja zmogljivosti. Peta rešitev (pri kateri se zmogljivost v nekih omejenih mejah spreminja na dnevni ravni) je tudi edina, za katero veljajo pogoji iz poglavja 3.1



Slika 5.4: Vse smiselne rešitve pri učenju modela brez zadnjega testiranja (rdeča pika) za kolesarja 3.

Rešitve pri kolesarju 5

Pri napovedih za kolesarja 5 (slika 5.5) vse napovedi opisujejo spreminjanje zmogljivosti, kot bi lahko izgledalo v resnici. Na podlagi predznanja za nobeno napoved ne moremo trditi, da je bolj verjetno pravilna kot ostale. Če upoštevamo pogoje iz poglavja 3.1, še vedno dobimo 3 rešitve (na sliki 5.6 in v tabeli 5.2).

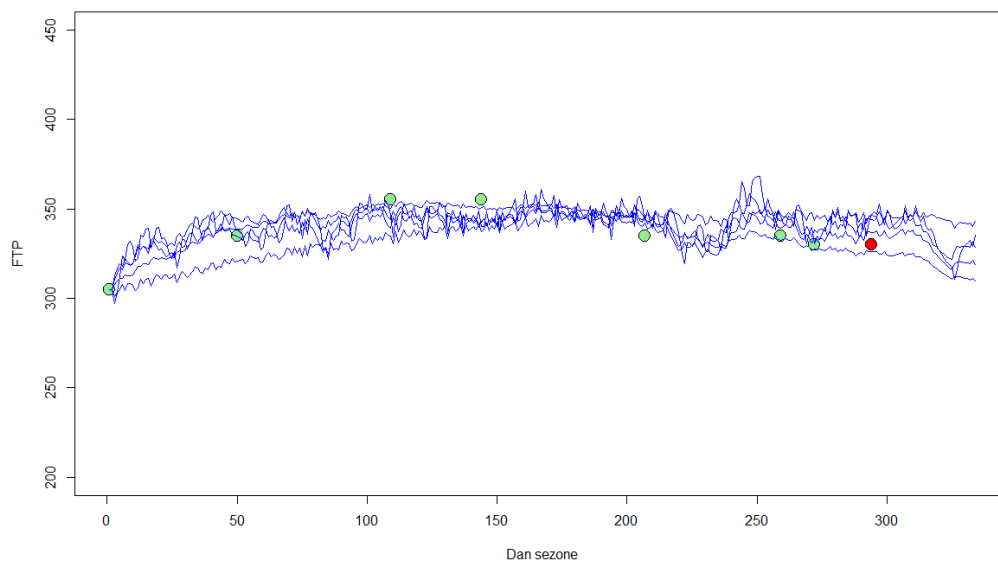
k_a	k_f	τ_a	τ_f	Vsota kvadratov napak	Napaka napovedi
80.11	80.09	83.06	83.07	163	3.9
25.03	24.98	10.01	10.02	649	13.8
25.18	25.15	33.08	33.12	420	6.8

Tabela 5.2: Smiselne rešitve za kolesarja 5, ki ustrezajo pogojem iz poglavja 3.1, vsota kvadratov napak na učni množici in napaka napovedi zadnjega testiranja.

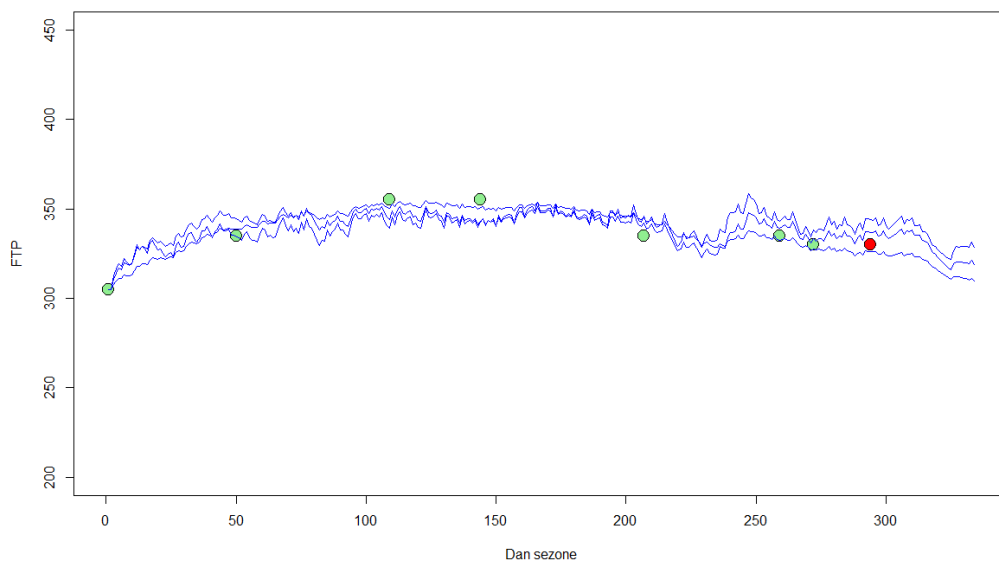
Opazimo, da se parametri različnih rešitev zelo razlikujejo, kljub temu, da so si napovedi podobne. To je precejšen problem tega modela, pogosto omenjen tudi v literaturi. Še posebej pri parametrih k_a in k_f , ker z njima

samo množimo obe komponenti (glejte enačbo 3.1). Če sta si podobna (pri vseh kolesarjih smo dobili le take rešitve), na napovedi ni skoraj nobene razlike, če sta vrednosti obeh blizu npr. 10 ali 50.

Tri preostale rešitve lahko primerjamo še po vsoti kvadratov napak na učni množici. Rešitev z najmanjšo napako na učni množici doseže najmanjšo napako napovedi na testni množici. Tudi srednja in najslabša rešitev (po napaki na učni množici) dosežeta srednjo in najslabšo napoved zadnjega testiranja. Želeli bomo torej rešiti problem lokalnih minimumov in najti rešitev, ki ima na učni množici najmanjšo vsoto napak.

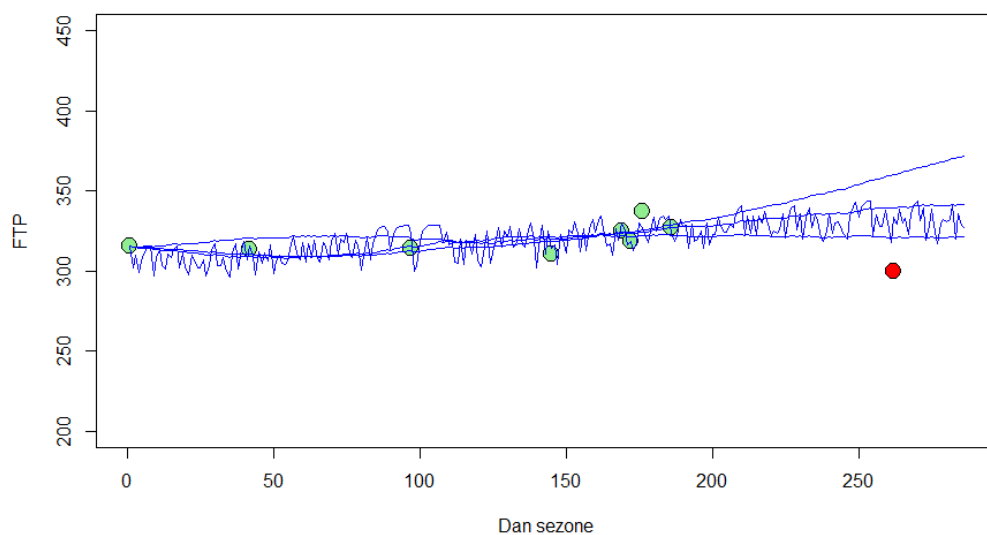


Slika 5.5: Vse smiselne rešitve pri učenju modela brez zadnjega testiranja (rdeča pika) za kolesarja 5.



Slika 5.6: Vse smiselne rešitve, ki ustrezajo pogojem iz poglavja 3.1, pri učenju modela brez zadnjega testiranja (rdeča pika) za kolesarja 5.

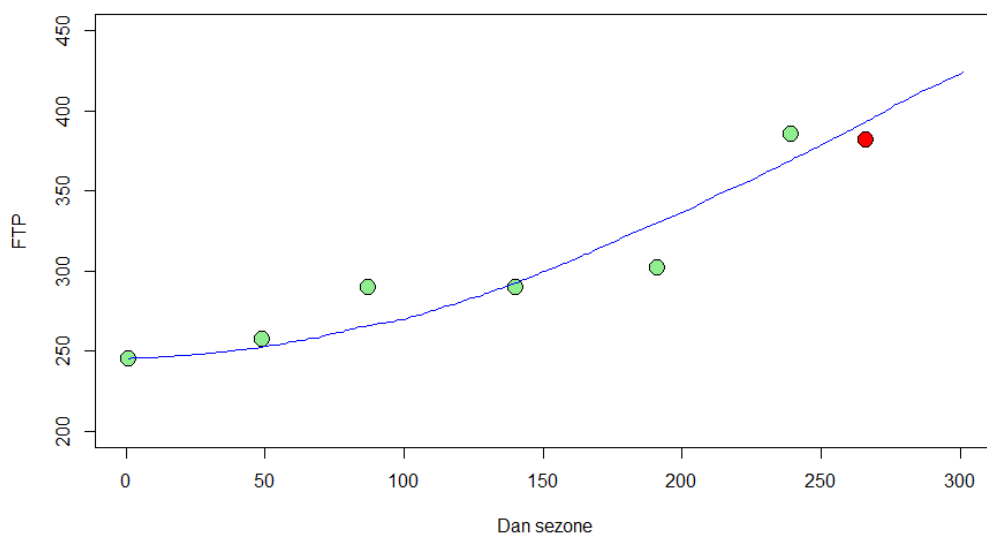
Rešitve pri kolesarju 6



Slika 5.7: Vse smiselne rešitve pri učenju modela brez zadnjega testiranja (rdeča pika) za kolesarja 6.

Enako kot pri kolesarju 3 (glejte poglavje 5.4) dobimo rešitve, pri katerih se zmogljivost spreminja zelo počasi, ki gotovo ne opišejo realne slike. Edina rešitev, ki bi lahko bila pravilna, je spet edina, ki ustreza pogojem iz poglavja 3.1. Prav tako nobena od rešitev niti približno ne napove rezultata zadnjega testiranja. Edini primer dobre napovedi je pri kolesarju 5 (glejte sliko 5.6), kjer pa je čas med zadnjim in predzadnjim testiranjem veliko krajši. Že zaradi tega je napoved bolj točna, saj napoveduje zmogljivost samo za nekaj dni naprej, pri kolesarjih 3 in 6 pa je razlika med zadnjim in predzadnjim testiranjem nekaj mesecev.

Rešitve pri kolesarju 2



Slika 5.8: Vse smiselne rešitve pri učenju modela brez zadnjega testiranja (rdeča pika) za kolesarja 2.

Pri kolesarju 2 s pogojem za smiselne rešitve, da je povprečna napaka napovedi testiranja manjša od 5 W, ne dobimo nobene rešitve. Preizkusimo manj strog pogoj, kjer je povprečna napaka napovedi testiranja lahko največ

7 W. Tako dobimo eno rešitev, ki se ne prilega niti učni množici, spreminjanje zmogljivosti, kot ga opiše, pa tudi ni mogoče. Model spreminjanja zmogljivosti tega kolesarja očitno ne more opisati.

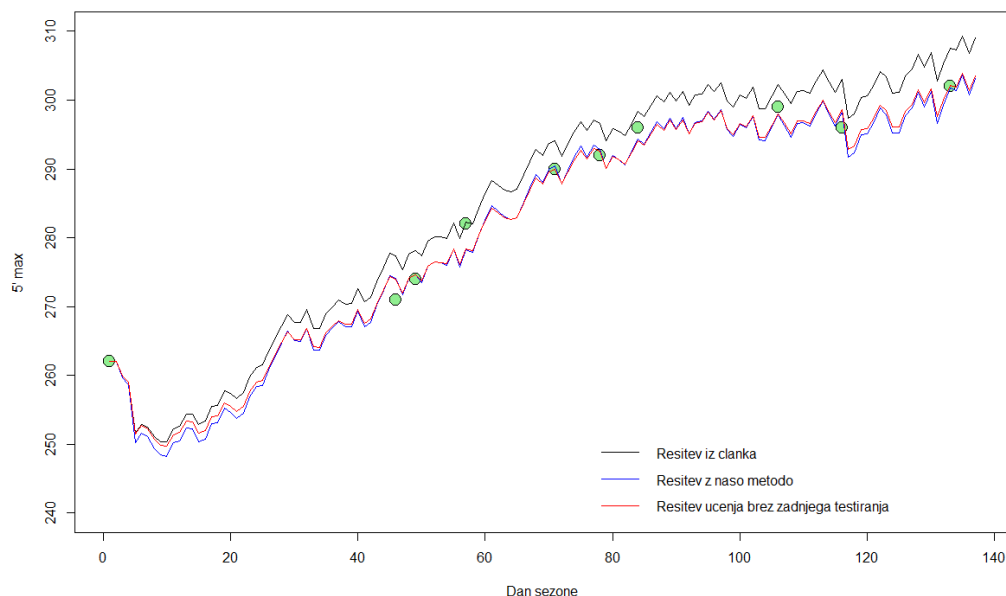
5.3 Preizkus modela na podatkih iz članka

V enem od člankov [5] testirajo uporabnost modela na podatkih kolesarja, ki je treninge opravljal na sobnem kolesu v obdobju 5 mesecev. Napor treningov je izražen z BikeScore, metodo ocenjevanja napora treningov, ki je zelo podobna metodi TSS, zmogljivost pa merijo s 5-minutnim testom maksimalne moči. Kvaliteto napovedi ocenijo z R^2 , ki je pričakovano zelo visok (0.979), saj model prilagajajo vsem rezultatom testiranj (teh je 9).

Najprej preizkusimo naš postopek minimizacije funkcije napake na podatkih vseh testiranj. V članku so prav tako minimizirali vsoto kvadratov napak, z Excelovim orodjem "Reševalnik", ki uporablja metodo generaliziranega gradientnega spusta [11]. Njihova rešitev minimizacije vsote kvadratov napak je $k_a = 0.18, k_f = 0.23, \tau_a = 36, \tau_f = 18$, katere vsota napak je 193. Z našo metodo pridemo do rešitve $k_a = 4.50, k_f = 4.56, \tau_a = 28.47, \tau_f = 27.84$, ki ima vsoto kvadratov napak 34. Napovedi modela, ki jih dobimo s tema rešitvama, so na sliki 5.9.

Nato poskusimo napovedati rezultat zadnjega testiranja na podatkih prejšnjih, model torej učimo na vseh testiranjih, razen zadnjem. Dobimo rešitev $k_a = 4.94, k_f = 5.00, \tau_a = 27.47, \tau_f = 26.88$, katere parametri so podobni rešitvi pri učenju modela na vseh testiranjih, napovedi pa sta praktično identični (glejte sliko 5.9).

Napoved modela zadnjega testiranja, ko model učimo na prejšnjih testiranjih, je v tem primeru točna. Možno je, da je napoved dobra predvsem zaradi kratkega časa med zadnjim in predzadnjim testiranjem (17 dni). Na kvaliteto napovedi vpliva tudi izbira metod ocenjevanja napora treningov in merjenja zmogljivosti, ki sta drugačni, kot pri naših podatkih. Pri kolesarjenju na sobnem kolesu je tudi veliko manj vpliva zunanjih dejavnikov, kot pri kolesarjenju zunaj.



Slika 5.9: Napoved spreminjanja zmogljivosti iz članka (črna), napoved z našim postopkom minimizacije (modra) in napoved, ki jo dobimo pri učenju modela brez zadnjega testiranja (rdeča).

5.4 Rezultati

5.4.1 Upoštevanje omejitev modela

V analizi rešitev (v poglavju 5.2) pri podatkih kolesarjev 3 in 6 opazimo, da se z upoštevanjem pogojev iz poglavja 3.1 znebimo napačnih rešitev, pri katerih se zmogljivost spreminja zelo počasi. Pri kolesarju 5 z upoštevanjem istih pogojev izločimo dve od petih rešitev, vseh pet pa opisuje spreminjanje zmogljivosti, kot bi lahko izgledalo v resnici. Pri optimizaciji je očitno smiselno upoštevati pogoje iz poglavja 3.1, saj izločimo nekaj napačnih rešitev, ne izgubimo pa nobene prave. V poglavju 4.1.1 smo opisali, kako jih lahko upoštevamo pri uporabi metode L-BFGS-B.

5.4.2 Napake na učni in testni množici

Kolesar 5 (glejte podpoglavje 5.2.1) je edini, pri katerem imamo kljub upoštevanju omejitev parametrov še vedno več kot eno rešitev. Če primerjamo napake napovedi na učni in testni množici (glejte tabelo 5.2) so med sabo korelirane, napoved z najmanjšo napako na učni množici doseže tudi najboljšo napoved zadnjega testiranja. Ker imamo samo en tak primer, ne moremo predpostavljati, da to velja tudi v splošnem. Vseeno pri vsaki minimizaciji iščemo rešitev z najmanjšo napako na učni množici, saj nismo našli nobenih rešitev z sicer višjo napako na učni množici, ki bi dosegle boljši rezultat na testni množici.

5.4.3 Kvaliteta napovedi

Ker smo pri analizi rešitev podatke že razdelili na učno in testno množico, lahko ocenimo kvaliteto napovedi modela. Pri kolesarju 2 se model ne prilega niti učni množici, ne najde nobene dobre rešitve. Pri kolesarjih 3 in 6 se model lepo prilega učni množici in z upoštevanjem omejitev parametrov lahko določimo eno rešitev, za katero menimo, da je najboljša. Vseeno pa nobena ne doseže dobre napovedi zadnjega testiranja, napovesta celo nasprotno spremembo. Le pri kolesarju 5 je napoved precej točna, ampak predvsem zaradi majhne časovne razlike med zadnjim in predzadnjim testiranjem. Problem pri tem kolesarju je tudi dejstvo, da imamo 3 rešitve, ki se zelo dobro prilegajo učni množici, kljub zelo različnim vrednostim parametrov pa dajo skoraj isto napoved.

5.4.4 Postopek minimizacije

Pri minimizaciji torej upoštevamo omejitve med parametri (glejte poglavje 4.1.1). Da se izognemo lokalnim minimumom je potrebno še dobro izbrati začetni približek rešitve. Najbolj preprost pristop je, da izračunamo vsoto kvadratov napak za več začetnih približkov in uporabimo tistega, ki ima najmanjšo vsoto kvadratov napak. Malo boljši pristop je uporaba metode simu-

liranega ohlajanja, ki učinkovito aproksimira globalni minimum kompleksnih funkcij. Metoda se premika med naključno generiranimi stanji (rešitvami), pri čemer se iz trenutnega stanja vedno premakne v boljšega, v slabšega pa z verjetnostjo, ki je odvisna od trenutne "temperature", ki z izvajanjem pada [10]. Tako dobimo približek globalnega minimuma, ki ga uporabimo kot začetni približek za metodo L-BFGS-B.

Obstaja še veliko metod, ki bi jih lahko uporabili za izbiro začetnega približka, metode razveji in omeji, genetski algoritmi, hevristične preiskovalne metode... Vseeno predlagamo uporabo metode simuliranega ohlajanja, saj je ena najpreprostejših metod in na naših podatkih doseže dobre rezultate.

Poglavje 6

Sklepne ugotovitve

Na naših podatkih z modelom nismo uspeli napovedati spreminjanja zmogljivosti. Model se v večini primerov dobro prilega učni množici, rezultati na testni množici pa so slabi. Zdi se, kot da se lahko prilega nekemu poljubnemu spreminjanje ciljne spremenljivke (že zaradi kompleksnosti - 4 parametri), nima pa nekih specifičnih lastnosti, zaradi katerih bi lahko uspešno napovedoval rezultate prihodnjih testiranj in s tem spreminjanje športne zmogljivosti.

6.1 Primerjava z ostalimi študijami

Morda je pri naših podatkih problem v načinu ocenjevanja napora treningov s TSS ali v ocenjevanju zmogljivosti s FTP, zato primerjamo naše ugotovitve z rezultati nekaterih ostalih študij. Pfeiffer [13] je pregledal 19 študij uporabnosti Banisterjevega modela, v katerih so na podatkih različnih športov analizirali kvaliteto napovedi. V študijah so uporabljeni različni pristopi k ocenjevanju napora treningov in testiranja zmogljivosti, pri teku in kolesarstvu večinoma uporabljajo oceno napora treningov TRIMP in napovedujejo vrednost *maksimalne aerobne kapacitete* ($VO_2 \text{ max}$). V vsaki izmed njih analizirajo podatke od 2 do 8 športnikov, kvaliteto napovedi pa ocenjujejo z R^2 .

R^2 oz. koeficient determinacije meri kvaliteto napovedi, po vrednosti je omejen med 0 do 1, razlagamo pa si ga lahko kot delež variance ciljne spremenljivke, ki smo ga z modelom uspeli pojasniti.

Pri vseh analiziranih študijah je R^2 visok, napoved se torej dobro prilega testiranjem zmogljivosti. Samo na podlagi visokega R^2 še ne moremo sklepati, da so napovedi modela dobre, vsak kompleksen model (npr. že polinom dovolj visoke stopnje) se bo dobro prilegal poljubnim vrednostim ciljne spremenljivke. Napoved rezultatov prihodnjih testiranj (in testiranj v testni množici) pa je vseeno lahko slaba, do česar je prišlo tudi pri naših podatkih (glejte poglavje 5.2.1).

Slabo kvaliteto napovedi na testni množici so ugotovili tudi v nekaterih ostalih študijah [7, 16], poleg tega pa kritizirajo tudi visoko korelacijo parametrov k_a in k_f .

6.2 Možnosti za nadaljnje delo

Napovedi modela so premalo natančne, da bi si z njimi lahko pomagali pri načrtovanju treningov [1]. Banisterjev model v nespremenjeni obliki obstaja že od leta 1976 [4], predlaganih je bilo nekaj alternativnih modelov, ki pa se niso izkazali za boljše [12]. Napovedi takih modelov bi lahko poizkusili izboljšati z uporabo bayesovskih metod, pri čemer bi za prilagajanje modela upoštevali podatke več športnikov. Na večji količini podatkov bi lahko tudi analitično določili, koliko rezultatov testiranj je potrebnih za stabilno napoved modela. Kot glavne pomanjkljivosti vseh teh modelov se izpostavlja neupoštevanje predznanja [16], po našem mnenju pa je velik problem upoštevanje le ene številske ocene vsakega treninga.

Dostopnost in količina podatkov o kolesarskih treningih je veliko boljša kot leta 1976, lahko npr. določimo, koliko časa je kolesar vozil s katero od različnih stopenj intenzivnosti. Za vsako stopnjo intenzivnosti vemo, kakšne fiziološke prilagoditve povzročajo, za vsako prilagoditev pa približno, kako vpliva na zmogljivost. Predlagamo raziskovalno delo na razvoju modela, ki bi razlikoval med različnimi stopnjami intenzivnosti in napovedoval zmogljivost z uporabo tega predznanja.

Literatura

- [1] T. Busso in T. Luc, “Using mathematical modeling in training planning,” *International Journal of Sports Physiology and Performance*, zv. 1, št. 4, str. 400–405, 2006.
- [2] T. Busso, C. Carasso, in J.-R. Lacour, “Adequacy of a systems structure in the modeling of training effects on performance,” *Journal of Applied Physiology*, zv. 71, št. 5, str. 2044–2049, 1991.
- [3] R. H. Byrd, P. Lu, J. Nocedal, in C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal on Scientific Computing*, zv. 16, št. 5, str. 1190–1208, 1995.
- [4] T. W. Calvert, E. W. Banister, M. V. Savage, in T. Bach, “A systems model of the effects of training on physical performance,” *IEEE Transactions on Systems, Man, and Cybernetics*, št. 2, str. 94–102, 1976.
- [5] D. C. Clarke in P. F. Skiba, “Rationale and resources for teaching the mathematical modeling of athletic training and performance,” *Advances in Physiology Education*, zv. 37, št. 2, str. 134–152, 2013.
- [6] J. Friel, *The cyclist’s training bible*, 3. izdaja. Velopress, 2003.
- [7] P. Hellard, M. Avalos, L. Lacoste, F. Barale, J.-C. Chatard, in G. P. Millet, “Assessing the limitations of the Banister model in monitoring training,” *Journal of sports sciences*, zv. 24, št. 05, str. 509–520, 2006.

- [8] A. Hunter in A. Coggan, *Training and racing with a powermeter*, 2. izdaja. Velopress, 2010.
- [9] M. Kavaš, *Trening kolesarjev : praktični vidik*, 1. izdaja. Samozaložba, 2013.
- [10] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simulated annealing," *Science*, zv. 220, št. 4598, str. 671–680, 1983.
- [11] L. S. Lasdon, R. L. Fox, in M. W. Ratner, "Nonlinear optimization using the generalized reduced gradient method," *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle*, zv. 8, št. 3, str. 73–103, 1974.
- [12] J. Perl, "Perpot: A metamodel for simulation of load performance interaction," *European Journal of Sport Science*, zv. 1, št. 2, str. 1–13, 2001.
- [13] M. Pfeiffer, "Modeling the relationship between training and performance—a comparison of two antagonistic concepts," *International journal of computer science in sport*, zv. 7, št. 2, str. 13–32, 2008.
- [14] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016. Dostopno na: <http://www.R-project.org/>
- [15] Strava. (2016) Strava segment - zelena magistrala climb. Dostopno na: <https://www.strava.com/segments/919815>
- [16] T. Taha in S. G. Thomas, "Systems modelling of the relationship between training and performance," *Sports Medicine*, zv. 33, št. 14, str. 1061–1073, 2003.
- [17] TrainingPeaks. (2016) Wko4 - trainingpeaks. Dostopno na: <https://www.trainingpeaks.com/wko4.html>