### UNIVERZITA KOMENSKÉHO V BRATISLAVE FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



# MINIMAXNÉ OPTIMÁLNE NÁVRHY REGRESNÝCH EXPERIMENTOV

DIPLOMOVÁ PRÁCA

Bc. Gabriel GROMAN

### UNIVERZITA KOMENSKÉHO V BRATISLAVE FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

# MINIMAXNÉ OPTIMÁLNE NÁVRHY REGRESNÝCH EXPERIMENTOV

### DIPLOMOVÁ PRÁCA

Študijný program:	Ekonomická a finančná matematika
Študijný odbor:	1114 Aplikovaná matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	prof. RNDr. Andrej Pázman, DrSc.

Bratislava 2014

Bc. Gabriel GROMAN





Univerzita Komenského v Bratislave Fakulta matematiky, fyziky a informatiky

## ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Študijný program:		Bc. Gabriel Groman ekonomická a finančná matematika (Jednoodborové žtódium, magistarský II. st., douná forma)		
Študijný od	dbor:	9.1.9. aplikovaná matematika		
Typ záverečnej práce: Jazyk záverečnej práce:		diplomová slovenský		
Názov:	Minimaxné opt	timálne návrhy regresných experimentov		
Ciel':	Minimaxné na problematika. oboznámiť sa	vrhovanie optimálnych experimentov je aktuálna štatistická Cieľom práce je, v nadväznosti na existujúce prednášky, v literatúre s niektorými existujúcimi prístupmi a preveriť		

	1	1	<b>J</b> )		5	1 2	
	oboznámiť sa v	literatúre s	niektorými	existujúcimi	prístupmi	a preveri	ť
	na počítači efektív	nosť nedávi	no publikova	anej metódy N	lyquista na	niektorýc	h
	konkrétnych ulohá	ich.	-			-	
V. 14		A 1 'D'	DC				

veduci:	prof. KNDr. Andrej Pazman, DrSc.
Katedra:	FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry:	prof. RNDr. Daniel Ševčovič, CSc.
Dátum zadania:	25.01.2013

Dátum schválenia: 04.02.2013

prof. RNDr. Daniel Ševčovič, CSc. garant študijného programu

študent

vedúci práce

## Poďakovanie

Úprimne ďakujem predovšetkým vedúcemu diplomovej práce prof. RNDr. Andrejovi Pázmanovi, DrSc. za odborné vedenie, cenné rady, trpezlivosť a čas, ktoré mi pri písaní tejto práce poskytol. Ďakujem aj Mgr. Lenke Filovej, PhD.. Veľká vďaka patrí mojim rodičom a priateľke za podporu počas celého môjho štúdia.

### Abstrakt

GROMAN, Gabriel: *Minimaxné optimálne návrhy regresných experimentov*. [Diplomová práca] - Univerzita Komenského v Bratislave. Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej matematiky a štatistiky. - Vedúci diplomovej práce: prof. RNDr. Andrej Pázman, DrSc. - Bratislava: FMFI UK, 2014, 79 s.

Diplomová práca sa zaoberá minimaxnými optimálnymi návrhmi regresných experimentov. Cieľom práce je, v nadväznosti na existujúce prednášky, oboznámiť sa v literatúre s niektorými existujúcimi prístupmi a preveriť nedávno publikovanú metódu Nyquista na konkrétnych úlohách. Teoretická časť práce sa venuje samotnej teórii navrhovania experimentov, lokálnym kritériám optimality, kritériám optimality v nelineárnom modeli a nutným a postačujúcim podmienkam optimality. Ďalej je podrobne rozpracovaná teória numerických metód hľadania optimálnych návrhov pre AVE kritérium, minimaxné kritérium a entropijnú regularizáciu minimaxného kritéria. Praktická časť práce obsahuje riešenie konkrétnych príkladov pomocou dvoch rôznych prístupov a analýzu výsledkov.

**Kľúčové slová:** minimaxné optimálne návrhy, AVE algoritmus, H-algoritmus, entropijná regularizácia

### Abstract

GROMAN, Gabriel: *Minimax optimal designs of regression experiments*. [Master Thesis] - Comenius University in Bratislava. Faculty of Mathematics, Physics and Informatics; Department of Applied Mathematics and Statistics. - Supervisor: prof. RNDr. Andrej Pázman, DrSc. - Bratislava: FMFI UK, 2014, 79 p.

Master thesis deals with the minimax optimal designs of regression experiments. The aim of the thesis is to become familiar by following the existing lectures with the existing approaches in literature and verify the recently published method by Nyquist in the specific cases. Theoretical part is dedicated to the theory how to design experiments, describes local optimality criteria, optimality criteria in nonlinear model and necessary and sufficient optimality conditions. Further on the theory of numeric methods to search optimal designs for AVE criteria, minimax criteria and entropy regularization of minimax criteria is elaborated in details. Practical part contains solutions for the specific cases based on two diverse approaches used together with the results analysis.

**Keywords:** minimax optimal designs, AVE algorithm, H-algorithm, entropy regularization

# Obsah

Zo	oznar	n obrá	zkov	9
Zo	oznar	n tabu	lliek	10
Zo	oznar	n použ	žitých symbolov	12
Ú	vod			13
1	Opt	imálne	e navrhovanie experimentov v nelineárnych modeloch	<b>14</b>
	1.1	Záklao	dy nelineárneho modelu, lokálne kritérium optimality	14
	1.2	Nutné	a postačujúce podmienky lokálnej optimality	15
	1.3	Klasic	ké lokálne kriteriálne funkcie a ich vlastnosti	16
	1.4	Kritér	ia optimality v nelineárnom modeli	17
		1.4.1	AVE (priemerovacie) kritérium optimality	17
		1.4.2	Minimaxné kritérium optimality	19
<b>2</b>	Nui	nerick	é metódy hľadania AVE optimálneho návrhu	<b>21</b>
3	Opi	s nume	erických metód hľadania minimaxného optimálneho návrhu	23
	3.1	H-algo	oritmus	23
		3.1.1	Podrobnejšie výsledky, keď množiny ${\mathscr X}$ a $\Theta$ sú konečné	26
	3.2	Entrop	pijná regularizácia minimaxného kritéria	27
4	Rie	šenie p	oríkladov	33
	4.1	Príkla	d 1	33
		4.1.1	Riešenie pomocou H-algoritmu [6]	34
		4.1.2	Riešenie pomocou entropijnej regularizácie	37
	4.2	Príkla	d 2	43
		4.2.1	Riešenie pomocou H-algoritmu [6]	45
		4.2.2	Riešenie pomocou entropijnej regularizácie	54
5	Dis	kusia l	a riešeným príkladom	58
Zá	iver			62

## Zoznam obrázkov

1	Pr. 1, entropijná regularizácia. Graf závislosti bodu návrhu od parametra $\lambda$	41
2	Pr. 1, entropijná regularizácia. Graf hodnôt kritérií v závislosti od pa-	
	rametra $\lambda$	42
3	Pr. 2, H-algoritmus. Body pridávané do návrhu $\xi^{\pi^{(1)}}$	50
4	Pr. 2, H-algoritmus. Body pridávané do návrhu $\xi^{\pi^{(2)}}$	51
5	Pr. 2, H-algoritmus. Body pridávané do návrhu $\xi^{\pi^{(3)}}$	53
6	Pr. 2, entropijná regularizácia. Graf hodnôt kritérií v závislosti od pa-	
	rametra $\lambda$	56
7	Pr. 2, entropijná regularizácia. Body pridávané do návrhu pre jednotlivé	
	hodnoty parametra $\lambda$	57

## Zoznam tabuliek

1	Pr. 1, H-algoritmus. Výsledky druhej iterácie	35
2	Pr. 1, H-algoritmus. Výsledky druhej iterácie podľa H. Nyquista	36
3	Pr. 1, H-algoritmus. Porovnanie našich a Nyquistových výsledkov druhej	
	iterácie	37
4	Pr. 1, entropijná regularizácia. Výsledky pre hodnotu parametra $\lambda=1$	39
5	Pr. 1, entropijná regularizácia. Výsledky pre hodnotu parametra $\lambda=0.1,2$	40
6	Pr. 1, entropijná regularizácia. Výsledky pre hodnotu parametra $\lambda=5,8$	40
7	Pr. 1, entropijná regularizácia. Záverečné porovnanie optimálnych ná-	
	vrhov pre rôzne hodnoty parametra $\lambda$	41
8	Pr. 2, H-algoritmus. Porovnanie výsledkov pre rozdelenie $\pi^{(1)}$	49
9	Pr. 2, H-algoritmus. Porovnanie výsledkov pre rozdelenie $\pi^{(2)}$	51
10	Pr. 2, H-algoritmus. Porovnanie výsledkov pre rozdelenie $\pi^{(3)}$	52
11	Pr. 2, entropijná regularizácia. Porovnanie výsledkov pre hodnotu para-	
	metra $\lambda = 0.1, 1$	55
12	Pr. 2, entropijná regularizácia. Porovnanie výsledkov pre hodnotu para-	
	metra $\lambda = 10,100$	56

# Zoznam použitých symbolov

OLS	ordinary least squares (metóda najmenších štvorcov)
$\mathscr{X}$	množina možných bodov merania v experimente
E[.]	stredná hodnota
Var[.]	variancia
$0 = (0_1, \dots, 0_m)^\top$	vektor núl
$oldsymbol{ heta} = ( heta_1, \dots,  heta_m)^ op$	vektor neznámych parametrov
Θ	množina všetkých neznámych parametrov
$\hat{\boldsymbol{ heta}} = (\hat{ heta}_1, \dots, \hat{ heta}_m)^ op$	vektor odhadov neznámych parametrov
П	množina všetkých pravdepodobnostných rozdelení na množine $\Theta$
$\mathcal{N}(oldsymbol{\mu},oldsymbol{\Sigma})$	normálne rozdelenie s kovariančnou maticou $\Sigma$ a strednou hodnotou $\mu$
$oldsymbol{M}(\xi,oldsymbol{ar{ heta}})$	informačná matica návrhu $\xi$ závislá od skutočnej hodnoty $\bar{\boldsymbol{\theta}}$ para-
	metra $\boldsymbol{ heta}$
$\mathbb{R}^{m  imes m}$	$m\times m$ rozmerný priestor matíc nad $\mathbb R$
$\Phi[oldsymbol{M}]$	kriteriálna funkcia
$ abla \Phi[oldsymbol{M}]$	gradient funkcie $\Phi$ v bod e $\boldsymbol{M}$
$\{ abla \Phi[oldsymbol{M}]\}_{ij}$	$ij\text{-ty}$ prvok gradientu funkci e $\Phi$ v bode $\boldsymbol{M}$
$\partial\Phi[ar{oldsymbol{M}},oldsymbol{M}]$	derivácia funkcie $\Phi$ v bod e $\bar{\boldsymbol{M}}$ a v smere $\boldsymbol{M}$
Ξ	množina všetkých návrhov experimentu
$\mathcal{M}$	množina všetkých informačných matíc v experimente
$tr(oldsymbol{M})$	stopa matice $M$
$\mathscr{M}({oldsymbol{M}})$	stĺpcový priestor matice $M$
$(\xi^\circ,\pi^\circ)$	sedlový bod typu minmax
$B(\xi,\pi)$	AVE kritérium optimality
π	apriórne rozdelenie
$\pi_0$	"najnepriaznivejšie rozdelenie"
$\xi^{\pi_0}$	AVE optimálny návrh vzhľadom k $\pi_0$
$B(\pi)$	$B(\xi^{\pi},\pi)$

$\Phi^m(\xi)$	minimaxné kritérium optimality
$\xi^m$	minimaxný optimálny návrh
$\Phi^e(\xi)$	entropijné kritérium optimality
$\xi^e$	entropijný optimálny návrh
$\alpha_n$	krok v algoritme
$\Delta_{\min}$	minimálna vzájomná vzdialenosť 2 bodov
$\omega_{ m min}$	minimálna váha bodu v návrhu
$\epsilon_{stop}$	maximálna "vzdialenosť" návrhu od optimálneho návrhu
$\epsilon^{\pi}$ (resp. $\epsilon_n$ )	"vzdialenosť" návrhu $\xi^\pi$ (resp. $\xi_n)$ od optimálneho návrhu
$\boldsymbol{\theta^{(l)}} \; ( ext{resp.} \; \boldsymbol{\theta^e})$	bod, v ktorom sa nadobúda ma x ${}_{\theta\in\Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(l)}},\boldsymbol{\theta})]$ (resp. $\Phi^m(\xi^e))$
$\delta^{(l)}$	Diracova miera v bode $\boldsymbol{\theta}^{(l)}$
$pi^{\pi^{(l)}}$ (resp. $pi^e$ )	počet iterácii pri konštrukcii návrhu $\xi^{\pi^{(l)}}$ (resp. $\xi^e)$
$\tau^{\pi^{(l)}}$ (resp. $\tau^e$ )	čas výpočtu pri konštrukcii návrhu $\xi^{\pi^{(l)}}$ (resp. $\xi^e)$
$w(x, \boldsymbol{\theta})$	$\frac{e^{\eta(x,\boldsymbol{\theta})}}{(1\!+\!e^{\eta(x,\boldsymbol{\theta})})^2}$

### Úvod

Jedna z najdôležitejších častí výskumu je fáza návrhu. Na to, aby sme mohli dôjsť k spoľahlivým záverom z fyzikálnych, biomedicínskych, farmaceutických a iných výskumov, musí návrh spĺňat určité požiadavky. Tieto požiadavky a samotná teória týchto oblastí je rozpracovaná v rôznych učebniciach a prácach. Jedna z málo docenených požiadaviek je dôležitosť dobre naplánovaného návrhu. Starostlivo naplánované návrhy môžu poskytnúť presné štatistické závery s minimálnymi nákladmi. Oblasť optimálnych návrhov experimentov spadá pod štatistiku. Poskytuje teóriu potrebnú na konštrukciu optimálnych, resp. približne optimálnych návrhov. Ako je uvedené v [11], existuje niekoľko monografií o tejto oblasti štatistiky, ktoré sa zaoberajú matematickým spracovaním problematiky. Niektoré príklady sú [2], [10] alebo [7]. V monografii [1] je mnoho príkladov z navrhovania experimentov v oblasti inžinierstva a farmácie.

Táto diplomová práca sa zaoberá minimaxnými optimálnymi návrhmi regresných experimentov. Cieľom práce je, v nadväznosti na existujúce prednášky [8], oboznámiť sa v literatúre s niektorými existujúcimi prístupmi a preveriť na počítači efektívnosť nedávno publikovanej metódy Nyquista na niektorých konkrétnych úlohách.

Diplomová práca je rozdelená do piatich kapitol. V prvej kapitole čitateľovi priblížime samotnú teóriu navrhovania optimálnych experimentov v nelineárnych modeloch. Kapitola 2 opisuje numerické metódy hľadania AVE optimálneho návrhu. V tretej kapitole opisujeme numerické metódy hľadania minimaxného optimálneho návrhu pomocou H-algoritmu a entropijnej regularizácie. V tejto časti taktiež podrobnejšie rozoberáme prípad, keď množiny  $\mathscr{X}$  a  $\Theta$  sú konečné. Nakoniec riešime konkrétne príklady oboma spomenutými prístupmi a v poslednej kapitole s názvom Diskusia k riešeným príkladom analyzujeme dosiahnuté výsledky.

# 1 Optimálne navrhovanie experimentov v nelineárnych modeloch

#### 1.1 Základy nelineárneho modelu, lokálne kritérium optimality

Nelineárna regresia vychádza z modelu

$$y_x = \eta(x, \theta) + \varepsilon_x,$$

kde  $E[\varepsilon_x] = 0$  a  $Var[\varepsilon_x] = \sigma^2$ . Narozdiel od lineárneho modelu, presnosť odhadu  $\hat{\theta}_{OLS}$ (ďalej len  $\hat{\theta}$ ) závisí od skutočnej hodnoty  $\bar{\theta}$  parametra  $\theta$  a preto je pri navrhovaní experimentov v nelineárnom modeli potrebná apriórna vedomosť o skutočnej hodnote  $\bar{\theta}$ . V optimálnom navrhovaní experimentov je cieľom nájsť taký návrh, ktorý bude k optimálnemu návrhu čo najbližsie, ktorý bude "*skoro*" optimálny [8].

**Definícia 1.1.** Návrh je ľubovoľná diskrétna pravdepodobnostná miera  $\xi$  na  $\mathscr{X}$ , ktorej nosič  $\{x \in \mathscr{X} : \xi(x) > 0\}$  je konečný. (Pozri [8])

Štatisticky by sme mohli návrh  $\xi(x)$  intepretovať ako relatívny počet meraní v bode x. Pri formulovaní kritérií optimality sa vychádza z asymptotickej normality OLS, kde pre dostatočne veľké N platí

$$\hat{\boldsymbol{\theta}} \sim \mathcal{N}\left(\bar{\boldsymbol{\theta}}, \sigma^2 \frac{1}{N} \boldsymbol{M}^{-1}(\boldsymbol{\xi}, \bar{\boldsymbol{\theta}})\right).$$

Z asymptotickej normality je zrejmé, že informácie o presnosti odhadov nesie informačná matica, ktorú si zadefinujeme v nasledujúcej definícii.

**Definícia 1.2.** Informačná matica  $M(\xi, \bar{\theta})$  má tvar

$$\begin{split} \boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\bar{\theta}}) &= \sum_{\boldsymbol{x} \in \mathscr{X}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\bar{\theta}}) \boldsymbol{f}^{\top}(\boldsymbol{x}, \boldsymbol{\bar{\theta}}) \boldsymbol{\xi}(\boldsymbol{x}), \\ kde \quad \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) &= \frac{\partial \eta(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \end{split}$$

Informačnú maticu  $\mathbf{M}(\xi, \bar{\boldsymbol{\theta}})$  niekedy píšeme skrátene  $\mathbf{M}$ . (Pozri [8])

Informačná matica  $M(\xi, \bar{\theta})$  je závislá od skutočnej hodnoty  $\bar{\theta}$  parametra  $\theta$ . Označme množinu všetkých návrhov experimentu  $\Xi$  a množinu všetkých informačných matíc  $\mathcal{M}$ 

$$\Xi \equiv \{\xi : \xi \text{ je návrh na } \mathscr{X}\}$$
$$\mathcal{M} \equiv \{M(\xi, \overline{\theta}) : \xi \in \Xi\}.$$

 $\Xi$ aj  ${\cal M}$  sú konvexné.

Pri hľadaní optimálneho návrhu chceme, aby bol náš návrh čo najpresnejší. To znamená, že hľadáme návrh s čo najmenšou varianciou funkcie parametra, teda návrh s čo "najväčšou" informačnou maticou. Tu sa nám vynára otázka, čo znamená "veľká" informačná matica. Odpoveď na túto otázku nám dáva **lokálne kritérium optimality**. Za tým účelom treba zvoliť nejaký bod  $\theta^* \in \Theta$  o ktorom aprióri predpokladáme, že je blízko skutočnej hodnoty  $\bar{\theta}$ .

**Definicia 1.3.** Lokálne kritérium optimality je funkcia  $\Phi : \mathbf{M}(\xi, \theta^*) \in \mathcal{M} \mapsto \Phi[\mathbf{M}(\xi, \theta^*)] \in \mathbb{R}$  taká, že ak  $\mathbf{M}(\xi, \theta^*) \succcurlyeq \mathbf{M}(\eta, \theta^*)$  (teda  $\forall_{u \in \mathbb{R}^m} : \mathbf{u}^\top \mathbf{M}(\xi, \theta^*) \mathbf{u} \ge \mathbf{u}^\top \mathbf{M}(\eta, \theta^*) \mathbf{u}$ ), potom  $\Phi[\mathbf{M}(\xi, \theta^*)] \le \Phi[\mathbf{M}(\eta, \theta^*)]$ . [8]

Návrh  $\xi^*$  je lokálne  $\Phi$ -optimálny (v lokalite bodu  $\theta^*$ ) práve vtedy, keď

$$\Phi[\boldsymbol{M}(\xi^*,\boldsymbol{\theta}^*)] = \min_{\xi\in\Xi} \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta}^*)].$$

(Pozri [8])

#### 1.2 Nutné a postačujúce podmienky lokálnej optimality

V tejto podkapitole sa oboznámime s nutnými a postačujúcimi podmienkami lokálnej optimality, no ešte predtým čitateľovi objasníme pojmy gradient a smerová derivácia. Nech  $\Phi[\mathbf{M}]$  je diferencovateľná reálna funkcia na otvorenej podmnožine  $\mathbb{R}^{m \times m}$ . Gradient funkcie  $\Phi$  je matica  $\nabla \Phi$  typu  $m \times m$  v tvare [8]

$$\{\nabla \Phi[\mathbf{M}]\}_{ij} = \frac{\partial \Phi[\mathbf{M}]}{\partial M_{ij}}, i, j = 1, \dots, m.$$

Smerová derivácia funckie  $\Phi[\mathbf{M}]$  v bode  $\mathbf{M}^*$  a v smere  $\mathbf{M}$  je definovaná [8]

$$\partial \Phi[\mathbf{M}^*, \mathbf{M}] = \lim_{\beta \to 0^+} \frac{\Phi[(1-\beta)\mathbf{M}^* + \beta\mathbf{M}] - \Phi[\mathbf{M}^*]}{\beta}.$$

Platia nasledujúce vety. (Podrobnosti pozri [8])

Veta 1.4. Ak  $\Phi$  je konvexná na  $\mathcal{M}$ , tak smerová derivácia existuje (konečná alebo nekonečná) v každom bode  $\mathcal{M}^*$  a v každom smere  $\mathcal{M}$ .

Súvis medzi gradientom a smerovou deriváciou poskytuje Veta 1.5.

Veta 1.5. Nech existuje  $\nabla \Phi[\mathbf{M}]$  v bode  $\mathbf{M}^*$ . Potom

$$\partial \Phi[\mathbf{M}^*, \mathbf{M}] = tr[\nabla \Phi[\mathbf{M}^*](\mathbf{M} - \mathbf{M}^*)].$$

Veta 1.6. Nech  $\Phi$  je konvexné lokálne kritérium optimality. Potom návrh  $\xi^*$  je lokálne  $\Phi$ -optimálny (t.j.  $\Phi[\mathbf{M}(\xi^*, \boldsymbol{\theta}^*)] = \min_{\xi \in \Xi} \Phi[\mathbf{M}(\xi, \boldsymbol{\theta}^*)])$  práve vtedy, keď

$$\forall_{\boldsymbol{\xi}} \partial \Phi[\boldsymbol{M}(\boldsymbol{\xi}^*, \boldsymbol{\theta}^*), \boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta}^*)] \geq 0.$$

V praxi sa táto veta ťažko využíva, oveľa jednoduhšie sa pracuje s nasledujúcou vetou, ktorej predpokladom je existencia gradientu.

Veta 1.7. Nech  $\Phi$  je konvexné lokálne kritérium optimality a nech existuje gradient  $\nabla \Phi[\mathbf{M}]$  v bode  $\mathbf{M} = \mathbf{M}(\xi^*, \boldsymbol{\theta}^*)$  (ďalej len  $\nabla \Phi[\mathbf{M}(\xi^*, \boldsymbol{\theta}^*)]$ ) pre všetky  $\boldsymbol{\theta}^*$ . Potom návrh  $\xi^*$  je lokálne  $\Phi$ -optimálny (t.j.  $\Phi[\mathbf{M}(\xi^*, \boldsymbol{\theta}^*)] = \min_{\xi \in \Xi} \Phi[\mathbf{M}(\xi, \boldsymbol{\theta}^*)]$ ) práve vtedy, keď

$$\min_{x\in\mathscr{X}}\boldsymbol{f}^{\top}(x,\boldsymbol{\theta}^*)\nabla\Phi[\boldsymbol{M}(\xi^*,\boldsymbol{\theta}^*)]\boldsymbol{f}(x,\boldsymbol{\theta}^*) = c(\xi^*,\boldsymbol{\theta}^*) \equiv tr\{\boldsymbol{M}(\xi^*,\boldsymbol{\theta}^*)\nabla\Phi[\boldsymbol{M}(\xi^*,\boldsymbol{\theta}^*)]\}.$$

Pomocou nasledujúcej vety vieme určiť, ako ďaleko je návrh $\mu$ od optimálneho návrhu.

Veta 1.8. Nech  $\Phi$  je konvexné lokálne kritérium optimality a nech existuje gradient  $\nabla \Phi[\mathbf{M}(\mu, \boldsymbol{\theta}^*)]$  pre všetky  $\boldsymbol{\theta}^*$ . Potom pre ľubovoľný návrh  $\mu$  platí

$$|\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta}^*)] - \min_{\xi\in\Xi} \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta}^*)]| \le c(\mu,\boldsymbol{\theta}^*) - \min_{x\in\mathscr{X}} \boldsymbol{f}^\top(x,\boldsymbol{\theta}^*) \nabla \Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta}^*)] \boldsymbol{f}(x,\boldsymbol{\theta}^*).$$

#### 1.3 Klasické lokálne kriteriálne funkcie a ich vlastnosti

V tejto časti uvedieme najčastejšie používané lokálne kriteriálne funkcie spolu s ich vlastnosťami. (Podrobne rozpracované v [8])

**D-optimalita** 
$$\Phi[\mathbf{M}] = \begin{cases} -ln[det(\mathbf{M})], & \text{ak } \mathbf{M} \text{ je regulárna,} \\ +\infty, & \text{inak.} \end{cases}$$

- konvexná funkcia, dokonca rýdzo konvexná na množine kladne definitných matíc,
- D-optimálna informačná matica je jediná,
- funkcia D-optimality je spojitá,
- pre regulárnu maticu **M** má gradient tvar:  $\nabla \Phi[\mathbf{M}] = -\mathbf{M}^{-1}$ ,

• návrh  $\xi^*$  je D-optimálny  $\Leftrightarrow \max_{x \in \mathscr{X}} \boldsymbol{f}^{\top}(x, \boldsymbol{\theta}^*) \boldsymbol{M}^{-1}(\xi^*, \boldsymbol{\theta}^*) \boldsymbol{f}(x, \boldsymbol{\theta}^*) = m.$ 

$$\mathbf{A}\text{-optimalita} \quad \Phi[\mathbf{M}] = \begin{cases} \sum_{i=1}^{m} Var_{\xi}[\hat{\theta}_i] = tr\{\mathbf{M}^{-1}\}, & \text{ak } \mathbf{M} \text{ je regulárna}, \\ +\infty, & \text{inak.} \end{cases}$$

- konvexná funkcia,
- funkcia A-optimality je spojitá,
- pre regulárnu maticu **M** má gradient tvar:  $\nabla \Phi[\mathbf{M}] = -\mathbf{M}^{-2}$ .

C-optimalita (pre pevné h)  $\Phi[\mathbf{M}] = \begin{cases} \mathbf{h}^{\top} \mathbf{M}^{-} \mathbf{h}, & \text{ak } \mathbf{h} \in \mathscr{M}(\mathbf{M}), \\ +\infty, & \text{inak.} \end{cases}$ 

- konvexná funkcia,
- funkcia C-optimality je polospojitá zdola na  $\mathcal{M}$ ,
- pre regulárnu maticu **M** má gradient tvar:  $\nabla \Phi[\mathbf{M}] = -\mathbf{M}^{-1}\mathbf{h}\mathbf{h}^{\top}\mathbf{M}^{-1}$ .

**E-optimalita**  $\Phi[\mathbf{M}] = \begin{cases} \max_{\boldsymbol{h}:\|\boldsymbol{h}\|=1} Var_{\boldsymbol{\xi}}[\boldsymbol{h}^{\top}\hat{\boldsymbol{\theta}}] = \frac{1}{\lambda_{\min}[\mathbf{M}]}, & \text{ak } \mathbf{M} \text{ je regulárna,} \\ +\infty, & \text{inak.} \end{cases}$ 

- konvexná funkcia,
- funkcia E-optimality je spojitá.

#### 1.4 Kritéria optimality v nelineárnom modeli

Ako sme spomenuli na začiatku tejto kapitoly, informačná matica v nelineárnom modeli závisí od skutočnej hodnoty  $\bar{\theta}$  parametra  $\theta$ . Preto pri hľadaní optimálneho návrhu nemôžeme použiť len lokálne kritéria optimality, ale musíme túto závislosť nejako ošetriť, zohľadniť. V nasledujúcich podkapitolách ukážeme možné prístupy, ako sa s týtmo faktom možno vysporiadať.

#### 1.4.1 AVE (priemerovacie) kritérium optimality

AVE (z anglického average (priemer)) kritérium optimality vychádza z klasických lokálnych kritérií optimality a definujeme ho nasledovne. Definícia 1.9.

$$B(\xi,\pi) = \int_{\Theta} \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})] d\pi(\boldsymbol{\theta}) = E_{\pi} \{ \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})] \},\$$

kde  $\Phi$  je klasická kriteriálna funkcia optimality a  $\pi(.)$  predstavuje apriórnu pravdepodobnostnú mieru na množine  $\Theta$ , ktorá vyjadruje apriórne očakávanie o skutočnej hodnote  $\bar{\boldsymbol{\theta}}$  parametra  $\boldsymbol{\theta}$ .[8]

Veta 1.10. Ak  $\Phi[\mathbf{M}(\xi, \boldsymbol{\theta})]$  je konvexná funkcia v  $\xi$  pre každé  $\boldsymbol{\theta}$ , potom  $B(\xi, \pi)$  je konvexná funkcia v  $\xi$  a lineárna v  $\pi(.)./8$ 

 $B(\xi, \pi)$ nie je funkciou jedinej informačnej matici a preto si smerovú deriváciu definujeme v nasledujúcej definícii.

**Definícia 1.11.** Smerová derivácia funkcie B v bode  $\mu$  a v smere  $\xi$  je

$$\partial B(\mu,\xi;\pi) = \lim_{\beta \to 0^+} \frac{B[(1-\beta)\mu + \beta\xi,\pi] - B(\mu,\pi)}{\beta}$$

Veta 1.12. Ak existuje gradient  $\nabla \Phi[\mathbf{M}(\mu, \boldsymbol{\theta})]$  pre všetky  $\boldsymbol{\theta} \in \Theta$  a ak možno zameniť poradie stredovania  $E_{\pi}[.]$ a derivovania podľa  $\beta$ , potom derivácia funkcie B v bode  $\mu$  a v smere  $\xi$  je

$$\partial B(\mu,\xi;\pi) = \sum_{x \in \mathscr{X}} \xi(x) E_{\pi} \{ \boldsymbol{f}^{\top}(x,\boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})] \boldsymbol{f}(x,\boldsymbol{\theta}) \} - E_{\pi} \{ c(\mu,\boldsymbol{\theta}) \},$$

kde  $c(\mu, \theta) \equiv tr\{\boldsymbol{M}(\mu, \theta) \nabla \Phi[\boldsymbol{M}(\mu, \theta)]\} = \sum_{x \in \mathscr{X}} \mu(x) \boldsymbol{f}^{\top}(x, \theta) \nabla \Phi[\boldsymbol{M}(\mu, \theta)] \boldsymbol{f}(x, \theta).[8]$ 

Nutné a postačujúce podmienky optimality

Návrh  $\xi^{\pi}$  je B-optimálny (nazývame tiež AVE optimálny návrh) pri danej apriórnej pravdepodobnostnej miere  $\pi(.)$  práve vtedy, keď

$$B(\xi^{\pi}, \pi) = \min_{\xi \in \Xi} B(\xi, \pi).$$
(1)

Veta 1.13. Návrh  $\xi^{\pi}$  je B-optimálny práve vtedy, keď

$$\forall_{\xi\in\Xi} \quad \partial B(\xi^{\pi},\xi;\pi) \ge 0.$$

Veta 1.14. Nech existuje gradient  $\nabla \Phi[\mathbf{M}(\xi^{\pi}, \boldsymbol{\theta})]$  pre všetky  $\boldsymbol{\theta} \in \Theta$ . Potom návrh  $\xi^{\pi}$  je B-optimálny práve vtedy, keď

$$\min_{x \in \mathscr{X}} E_{\pi} \{ \boldsymbol{f}^{\top}(x, \boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi^{\pi}, \boldsymbol{\theta})] \boldsymbol{f}(x, \boldsymbol{\theta}) \} = E_{\pi} \{ c(\xi^{\pi}, \boldsymbol{\theta}) \} \equiv$$
$$\equiv \sum_{x \in \mathscr{X}} \xi^{\pi}(x) E_{\pi} \{ \boldsymbol{f}^{\top}(x, \boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi^{\pi}, \boldsymbol{\theta})] \boldsymbol{f}(x, \boldsymbol{\theta}) \}.$$

Nasledujúcou vetou vieme oceniť, ako ďaleko je návr<br/>h $\mu$ od AVE optimálneho návrhu.

Veta 1.15. Nech  $\Phi$  je konvexné lokálne kritérium optimality a nech existuje gradient  $\nabla \Phi[\mathbf{M}(\mu, \boldsymbol{\theta})]$  pre všetky  $\boldsymbol{\theta} \in \Theta$ . Potom pre ľubovoľný návrh  $\mu$  platí

$$|B(\mu,\pi) - \min_{\xi \in \Xi} B(\xi,\pi)| \le E_{\pi} \{ c(\mu,\theta) \} - \min_{x \in \mathscr{X}} E_{\pi} \{ \boldsymbol{f}^{\top}(x,\theta) \nabla \Phi[\boldsymbol{M}(\mu,\theta)] \boldsymbol{f}(x,\theta) \},\$$

kde  $E_{\pi}\{c(\mu, \theta)\} = \sum_{x \in \mathscr{X}} \mu(x) E_{\pi}\{f^{\top}(x, \theta) \nabla \Phi[M(\mu, \theta)]f(x, \theta)\}.$ 

(Pre bližšie nahliadnutie k predchádzajúcim definíciám a vetám pozri [8])

#### 1.4.2 Minimaxné kritérium optimality

Minimaxné kritérium takisto vychádza z klasických lokálnych kritérií optimality. Týmto kritériom hľadáme optimálny návrh  $\xi^m$  pri "*najhoršej*" možnej hodnote  $\theta^{\diamond}$  parametra  $\theta$ .

**Definícia 1.16.** [8] Majme klasické lokálne kritérium optimality  $\Phi$ . Minimaxné kritérium optimality je definované vzťahom

$$\Phi^m(\xi) = \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})].$$

Veta 1.17. Ak  $\Phi[\mathbf{M}(\xi, \boldsymbol{\theta})]$  je konvexná funkcia v  $\xi$  pre každé  $\boldsymbol{\theta} \in \Theta$ , potom  $\Phi^m(\xi)$  je konvexná funkcia. (Pozri [8])

#### Nutné a postačujúce podmienky optimality

Návrh $\xi^m$  je $\Phi^m$ -optimálny (nazývame tiež minimaxný optimálny návrh) práve vtedy, keď

$$\Phi^m(\xi^m) = \min_{\xi \in \Xi} \Phi^m(\xi).$$

**Veta 1.18.** Nech  $\xi^{\gamma}$  je taký návrh, že podmnožina množiny  $\Theta$  definovaná vzťahom

$$\mathscr{M} = \arg \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi^{\gamma}, \boldsymbol{\theta})]$$

je konečná.

1. Ak je návrh  $\xi^{\gamma}$  AVE optimálny pre apriórnu pravdepodobnosť  $\gamma$  takú, že

$$\sum_{\boldsymbol{\theta} \in \mathcal{M}} \gamma(\boldsymbol{\theta}) = 1,$$

potom návrh  $\xi^{\gamma}$  je  $\Phi^m$ -optimálny na  $\Theta$ .

2. Navyše, ak je množina  $\Theta$  kompaktná, tak platí aj obrátená implikácia.

Poznamenávame, že naše tvrdenie 1 Vety 1.18 je totožné s tvrdením (i) Vety 2.1 v článku [4]. V spomínanom článku je toto tvrdenie formulované takto

(i) Ak existuje 
$$\gamma : \mathscr{M} \to [0,1]$$
 s podmienkou  $\sum_{\theta \in \mathscr{M}} \gamma(\theta) = 1$  a

$$\forall_{\xi\in\Xi} \quad \sum_{\boldsymbol{\theta}\in\mathscr{M}} \gamma(\boldsymbol{\theta}) \partial \Phi[\boldsymbol{M}(\xi^{\gamma}, \boldsymbol{\theta}), \boldsymbol{M}(\xi, \boldsymbol{\theta})] \geq 0,$$

potom návrh  $\xi^{\gamma}$  je  $\Phi^m$ -optimálny.

(Pre bližšie informácie a podrobný dôkaz pozri [4])

# 2 Numerické metódy hľadania AVE optimálneho návrhu

Táto kapitola sa venuje algoritmu na hľadanie AVE optimálneho návrhu, teda návrhu s kritériom definovaným nasledovne

$$B(\xi,\pi) = \int_{\Theta} \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})] d\pi(\boldsymbol{\theta}) = E_{\pi} \{ \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})] \},\$$

kde  $\pi(\boldsymbol{\theta})$  je dané a hľadáme  $\xi^{\pi} = \arg\min_{\xi \in \Xi} B(\xi, \pi).$ 

kde

Samotný algoritmus pozostáva z 5 nasledujúcich krokov. Máme dané  $\mathscr{X}, \Theta, \pi(\boldsymbol{\theta})$ :

**Krok 1** V prvom kroku vytvoríme diskrétny počiatočný návrh  $\xi_0$  taký, ktorého informačná matica bude regulárna (t.j. det  $[\mathbf{M}(\xi_0, \boldsymbol{\theta})] \neq 0, \forall_{\boldsymbol{\theta} \in supp \ \pi(\boldsymbol{\theta})})$ 

$$\xi_0 = \begin{cases} x_1 & x_2 & \dots & x_q \\ \omega_1 & \omega_2 & \dots & \omega_q \end{cases}, \text{ kde } \sum_{i=1}^q \omega_i = 1.$$

Určíme  $\epsilon_{stop}$ , teda najväčšiu možnú "vzdialenosť" nášho návrhu od optimálneho návrhu, ktorú akceptujeme. Ako pravidlo zastavenia použijeme podmienku približnej optimality podľa Vety 1.15

$$|E_{\pi}\{c(\xi_{n},\boldsymbol{\theta})\} - \min_{x\in\mathscr{X}} E_{\pi}\{\boldsymbol{f}^{\top}(x,\boldsymbol{\theta})\nabla\Phi[\boldsymbol{M}(\xi_{n},\boldsymbol{\theta})]\boldsymbol{f}(x,\boldsymbol{\theta})\}| \leq \epsilon_{stop}, \qquad (2)$$
  
kde  $E_{\pi}\{c(\xi_{n},\boldsymbol{\theta})\} = \sum_{x\in\mathscr{X}} \xi_{n}(x)E_{\pi}\{\boldsymbol{f}^{\top}(x,\boldsymbol{\theta})\nabla\Phi[\boldsymbol{M}(\xi_{n},\boldsymbol{\theta})]\boldsymbol{f}(x,\boldsymbol{\theta})\}.$ 

Pre D-optimalitu platí  $E_{\pi}\{c(\xi_n, \boldsymbol{\theta})\} = m$ , teda výraz  $E_{\pi}\{c(\xi_n, \boldsymbol{\theta})\}$  má hodnotu rovnú počtu parametrov. Overíme, či náš počiatočný návrh spĺňa pravidlo zastavenia, ak áno, tak počiatočný návrh je približne AVE optimálny, ak nie, pokračujeme Krokom 2.

**Krok 2** Nájdeme bod  $x_{min} = \arg\min_{x \in \mathscr{X}} E_{\pi} \{ \boldsymbol{f}^{\top}(x, \boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi_n, \boldsymbol{\theta})] \boldsymbol{f}(x, \boldsymbol{\theta}) \}$ a zvolíme vhodnú dĺžku kroku  $\alpha_n$ . Nájdený bod  $x_{min}$  pridáme do nášho návrhu, čím vytvoríme nový návrh

$$\xi_{n+1} = (1 - \alpha_n)\xi_n + \alpha_n \delta_{x_{min}}(x),$$
  
$$\delta_{x_{min}}(.) \text{ je diskrétny jednobodový návrh, t.j. } \delta_{x_{min}}(x) = \begin{cases} 1, & \text{ak } x = x_{min}, \\ 0, & \text{ak } x \neq x_{min}. \end{cases}$$

- **Krok 3** Zvolíme  $\Delta_{min}$  minimálnu vzájomnú vzdialenosť 2 bodov  $x_i$  a  $x_j$ . Ak je vzájomná vzdialenosť bodov  $x_i$  a  $x_j$  menšia ako zvolené  $\Delta_{min}$ , tak tieto body zlúčime do jedného bodu  $x_k$ , ktorého váha  $\omega_k = \omega_i + \omega_j$ . Tento postup opakujeme až dovtedy, dokým vzájomná vzdialenosť každých dvoch bodov nie je väčšia ako  $\Delta_{min}$ . Po tomto kroku pokračujeme Krokom 4.
- **Krok 4** Zvolíme číslo  $\omega_{min} > 0$  a body, ktorých váha je menšia ako zvolená  $\omega_{min}$ z návrhu vyhodíme a váhy ostatných bodov prepočítame tak, aby v súčte dávali 1, čím vytvoríme **redukovaný** návrh  $\boldsymbol{\xi}_{n+1}^r$ .
- Krok 5 Overíme podmienku približnej optimality (2) definovanú v Kroku 1, ak je splnená, návrh je približne AVE optimálny. Ak návrh nespĺňa pravidlo zastavenia, tak sa s týmto návrhom vrátime do Kroku číslo 2.

Tento postup opakujeme, až kým nenájdeme **približne AVE optimálny návrh**, teda návrh, ktorý spĺňa pravidlo zastavenia (2). Chceme podotknúť, že spájanie a vymazávanie bodov, teda Krok 3 a 4, sa zväčša realizuje až po k-tej iterácii, kde sa číslo k určuje aposteriórne, teda zo skúseností. Takisto chceme dodať, že krok  $\alpha_n$  sa zväčša volí  $\frac{1}{q+1}$ , kde q je počet bodov v návrhu  $\xi_n$ . V tejto kapitole sme popísali všeobecný algoritmus na skonštruovanie približne AVE optimálneho návrhu. Každý príklad je však špecifický a preto je niekedy pri konštruovaní (približne) AVE optimálneho návrhu potrebné dodať ešte nejaké reštrikcie, ktoré zabezpečia konštrukciu "lepšieho" návrhu. Preto nie je možné popísať všeobecný algoritmus, ktorý je vhodný pre každý príklad.

V našom algoritme sa stretneme s rátaním inverzie informačnej matice (gradient Doptimality) a tak uvedieme užitočný trik, pomocou ktorého si výpočet zjednodušíme (pre jednoduchosť použijeme informačnú maticu v tvare  $M(\xi_n)$ ). (Pozri [8])

# 3 Opis numerických metód hľadania minimaxného optimálneho návrhu

#### 3.1 H-algoritmus

V tejto časti predstavíme a popíšeme H-algoritmus, ktorý vymyslel H. Nyquist a publikoval v práci [5]. Je to algoritmus na skonštruovanie minimaxného optimálneho návrhu, ktorý využíva vzťah medzi minimaxným optimálnym návrhom a AVE optimálnym návrhom pre "*najnepriaznivejšie rozdelenie*". Pred popisom samotného algoritmu predostrieme čitateľovi krátky úvod [6]. Budeme pracovať za nasledujúcich predpokladov.

 $A_1: x$  je q-rozmerný vektor,  $\mathscr{X}$  je kompaktná podmnožina  $\mathbb{R}^q, x \in \mathscr{X} \subset \mathbb{R}^q$ .

- $A_2: \Phi[\mathbf{M}(\xi, \boldsymbol{\theta})]$  je spojitá, rýdzokonvexná a diferencovateľná reálna funkcia pre všetky  $\boldsymbol{\theta}$ na množine  $m \times m$  symetrických matíc a je zdola ohraničená.
- $A_3: \boldsymbol{M}(\xi, \boldsymbol{\theta})$  je spojitá funkcia premennej  $\boldsymbol{\theta}$ .
- $A_4: \Theta$  je kompaktná množina.

**AVE optimálny návrh \xi^{\pi}**s ohľadom na apriórnu pravdepodobnostnú mieru  $\pi$  je definovaný v (1), teda platí

$$B(\xi^{\pi}, \pi) = \min_{\xi \in \Xi} B(\xi, \pi).$$
(3)

Pre zjednodušenie budeme používať označenie  $B(\xi^{\pi}, \pi) = B(\pi)$ . Predpoklady  $A_1$  a  $A_2$  zabezpečujú existenciu lokálnych a AVE optimálnych návrhov.

Minimaxný optimálny návrh $\xi^m$  je návrh, ktorý spĺňa nasledovné

$$\Phi^{m}(\xi^{m}) = \min_{\xi \in \Xi} \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})].$$
(4)

Definujeme  $\Pi$ ako množinu všetkých pravdepodobnostných rozdelení na množine $\Theta.$ Je možné ukázať, že

$$\max_{\boldsymbol{\theta}\in\Theta} \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})] = \max_{\pi\in\Pi} B(\xi,\pi)$$
(5)

a preto výraz (4) možno prepísať takto

$$\Phi^m(\xi^m) = \min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi).$$

Apriórne rozdelenie  $\pi_0$  s vlastnosťou

$$B(\xi^{\pi_0}, \pi_0) = \max_{\pi \in \Pi} B(\xi^{\pi}, \pi)$$

sa nazýva "najnepriaznivejšie rozdelenie" vzhľadom k Π. Zo vzťahu (3) vyplýva

$$B(\xi^{\pi_0}, \pi_0) = \max_{\pi \in \Pi} \min_{\xi \in \Xi} B(\xi, \pi).$$

Z [6] je prebratá nasledujúca veta.

**Veta 3.1.** Predpokladajme, že  $\xi^{\pi_0}$  AVE optimálny návrh vzhľadom k nejakému  $\pi_0 \in \Pi$ má vlastnosť

$$\Phi[\boldsymbol{M}(\xi^{\pi_0}, \boldsymbol{\theta})] \le B(\pi_0) \text{ pre všetky } \boldsymbol{\theta} \in \Theta.$$
(6)

Potom

- 1.  $\min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi) = \max_{\pi \in \Pi} \min_{\xi \in \Xi} B(\xi, \pi) = B(\pi_0),$
- 2.  $\pi_0$  je "najnepriaznivejšie rozdelenie" vzhľadom k  $\Pi$ ,
- 3.  $\xi^{\pi_0}$  je minimaxný optimálny návrh, teda  $\xi^{\pi_0} = \xi^m$ .

Poznamenávame, že nerovnosť (6) je vlastne pravidlo zastavenia pre ďalej uvedený algoritmus.

Body 1,2 a 3 veľmi súvisia s jedným z hlavných výsledkov teórie hier, ktorý hovorí, že keby množiny  $\mathscr{X}$  a  $\Theta$  boli konečné, tak potom množiny  $\Xi$  a  $\Pi$  sú kompaktné a z toho, že  $B(\xi, \pi)$  je konvexné v  $\xi$  a konkávne v  $\pi$  by vyplývali body 1,2 a 3. Prípadu konečnosti množín  $\mathscr{X}$  a  $\Theta$  sa bližšie venujeme v Podkapitole 3.1.1.

Dôkaz (podrobnejšie ako v [6]):

$$B(\xi,\pi) \ge \min_{\xi \in \Xi} B(\xi,\pi), \forall_{\xi \in \Xi}, \forall_{\pi \in \Pi},$$
(7)

keďže nerovnosť platí pre  $\forall_{\pi \in \Pi}$ , tak platí aj pre maximum

$$\max_{\pi \in \Pi} B(\xi, \pi) \ge \max_{\pi \in \Pi} \min_{\xi \in \Xi} B(\xi, \pi), \forall_{\xi \in \Xi}.$$
(8)

Predchádzajúci výraz platí pre  $\forall_{\xi \in \Xi}$ , tak platí aj pre minimum

$$\min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi) \ge \max_{\pi \in \Pi} \min_{\xi \in \Xi} B(\xi, \pi).$$
(9)

Na druhej strane máme

$$\min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi) \le \max_{\pi \in \Pi} B(\xi, \pi), \forall_{\xi \in \Xi}.$$
(10)

Nerovnosť platí pre  $\forall_{\xi \in \Xi}$ , teda aj pre AVE optimálny návrh  $\xi^{\pi_0}$ 

$$\min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi) \le \max_{\pi \in \Pi} B(\xi^{\pi_0}, \pi) = \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi_0}, \boldsymbol{\theta})].$$
(11)

Z rovnosti (5) a vlastnosti (6) dostávame

$$\min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi) \le \max_{\pi \in \Pi} B(\xi^{\pi_0}, \pi) =$$
(12)

$$= \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi_0}, \boldsymbol{\theta})] \le B(\pi_0) = \min_{\xi \in \Xi} B(\xi, \pi_0) \le \max_{\pi \in \Pi} \min_{\xi \in \Xi} B(\xi, \pi),$$
(13)

a preto v (9) a (13) musí byť rovnosť.

Pri samotnom skonštruovaní algoritmu začneme tak, že vygenerujeme počiatočné apriórne rozdelenie a nájdeme k tomu prislúchajúci (približne) AVE optimálny návrh. V každej nasledujúcej iterácii pozmeníme apriórne rozdelenie a opäť nájdeme (približne) AVE optimálny návrh. Takto pokračujeme, až kým nenájdeme *"najnepriaznivejšie rozdelenie"* a k tomu prislúchajúci (približne) AVE optimálny návrh, ktorý je zároveň (približne) minimaxným optimálnym návrhom. H-algoritmus krok po kroku opíšeme nasledovne, tak ako v [6].

Máme dané  $\mathscr{X}, \Theta$ :

**Krok 0** Nastavíme hodnotu l = 1 a formálne definujeme  $B(\pi^{(0)}) = -\infty$ . Definujeme pomocnú sieť bodov na intervale  $\langle 0, 1 \rangle$ :  $H = \{h_0, ..., h_T\}$ , kde  $0 = h_0 < h_1 < ... < h_T \leq 1$ . Zvolíme nejaké počiatočné apriórne rozdelenie  $\pi^{(1)}$ , skonštruujeme (približne) AVE optimálny návrh  $\xi^{\pi^{(1)}}$  a vyčíslime  $B(\pi^{(1)})$ .

Krok 1 Overíme, či

$$\Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(l)}},\boldsymbol{\theta})] \leq B(\pi^{(l)}) \text{ pre všetky } \boldsymbol{\theta} \in \Theta.$$

Ak je podmienka splnená, algoritmus zastavíme,  $\pi^{(l)}$  je "*najnepriaznivejšie rozdelenie"* a  $\xi^{\pi^{(l)}}$  je (približne) minimaxný optimálny návrh. Ak podmienka splnená nie je, tak pokračujeme Krokom 2. 3

**Krok 2** Nájdeme bod  $\boldsymbol{\theta}^{(l)} = \arg \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(l)}}, \boldsymbol{\theta})]$  a označíme  $\delta^{(l)}$  Diracovu mieru v bode  $\boldsymbol{\theta}^{(l)}$ .

**Krok 3** Generujeme *T* nových apriórnych rozdelení:  $\pi_t^{(l+1)} = (1 - h_t)\pi^{(l)} + h_t\delta^{(l)}$ . Pre každé nové apriórne rozdelenie nájdeme prislúchajúci (približne) AVE optimálny návrh  $\xi^{\pi_t^{(l+1)}}$  a vyčíslime  $B(\pi_t^{(l+1)}), t=1,...,T$ . Označme  $t_0 = \arg \max_{t \in \{1,...,T\}} B(\pi_t^{(l+1)})$ . Ďalej označíme  $\pi^{(l+1)} = \pi_{t_0}^{(l+1)}$ . Návrh  $\xi^{\pi^{(l+1)}}$  definujeme ako  $\xi^{\pi^{(l+1)}} = \xi^{\pi_{t_0}^{(l+1)}}$  (je to (približne) AVE optimálny návrh pre  $\pi^{(l+1)}$ ).

Krok 4 Overíme nasledujúce

$$B(\pi^{(l+1)}) \ge B(\pi^{(l)}).$$

Ak podmienka nie je splnená, tak definujeme hustejšiu sieť H a zopakujeme Krok číslo 3, ináč nastavíme l = l + 1 a vrátime sa do Kroku 1.

Tento algoritmus budeme preverovať na príkladoch. Napriek *Theorem 2* z [6] vetu o konvergencii algoritmu nepokladáme za preukázanú.

#### 3.1.1 Podrobnejšie výsledky, keď množiny ${\mathscr X}$ <br/>a $\Theta$ sú konečné

V tejto podkapitole ukážeme, ako za predpokladu konečnosti množín  $\mathscr{X}$  a  $\Theta$  možno dôjsť k bodom 1,2 a 3 Vety 3.1 bez využitia predpokladu (6), respektíve, že Veta 3.1 je v tomto prípade nutnou i postačujúcou podmienkou optimality.

**Tvrdenie 3.2.** Množiny  $\mathscr{X}$  a  $\Theta$  sú konečné  $\Rightarrow$  množiny  $\Xi$  a  $\Pi$  sú kompaktné a konvexné.

**Tvrdenie 3.3.**  $B(\xi, \pi)$  je spojitá funkcia premennej  $\xi$  a  $\pi$ , podľa  $\xi$  je konvexná a podľa  $\pi$  lineárna.

Veta 3.4. (Sion, 1905, [3])

Nech  $\Xi$  a  $\Pi$  sú konvexné a kompaktné množiny. Nech  $B(\xi, \pi)$  je podľa  $\xi$  zdola-polospojitá a kvázikonvexná a podľa  $\pi$  zhora-polospojitá a kvázikonkávna. Potom  $B(\xi, \pi)$  má vzhľadom na množinu  $\Xi \times \Pi$  sedlový bod typu minmax.

Táto veta hovorí o existencii sedlového bodu typu minmax. Nasledúca veta hovorí o vzťahu medzi sedlovým bodom minmax a kombinovanými extrémami.

**Veta 3.5.** (Veta o minmaxe, [3])

Funkcia  $B(\xi, \pi)$  definovaná na  $\Xi \times \Pi$  má sedlový bod  $(\xi^{\circ}, \pi^{\circ})$  typu minmax vzhľadom na  $\Xi \times \Pi$  práve vtedy, ak existujú výrazy

- $\min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi),$
- $\max_{\pi \in \Pi} \min_{\xi \in \Xi} B(\xi, \pi),$

a ich hodnoty sú rovnaké. Potom tiež platí

$$B(\xi^{\circ}, \pi^{\circ}) = \min_{\xi \in \Xi} \max_{\pi \in \Pi} B(\xi, \pi) = \max_{\pi \in \Pi} \min_{\xi \in \Xi} B(\xi, \pi).$$

Vetou 3.4 sme ukázali, že funkcia  $B(\xi, \pi)$  má sedlový bod typu minmax. Veta o minmaxe v našom prípade jasne hovorí o existencii a rovnosti výrazov min max  $B(\xi, \pi)$  a max min  $B(\xi, \pi)$ . Ukázali sme, že táto rovnosť, čiže bod 1 Vety 3.1 v tomto prípade platí aj bez predpokladu (6). Body 2 a 3 Vety 3.1 vyplývajú z nasledujúcej vety.

Veta 3.6. Nech  $\xi^m$  je minimaxný návrh, potom na množine  $\mathcal{M}$ , ktorá je podľa Vety 1.18 definovaná

$$\mathscr{M} = \arg \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi^m, \boldsymbol{\theta})]$$

existuje  $\pi_0$  také, že  $\xi^m$  je AVE optimálny návrh vzhľadom na  $\pi_0$  a platí pravidlo zastavenia (6).

#### 3.2 Entropijná regularizácia minimaxného kritéria

Pomocou entropie možno aproximovať minimaxné kritérium na špeciálny druh AVE kritéria (pozri [9] kap. 8.3.2). Z prvej kapitoly vieme, že minimaxné kritérium je definované nasledovne

$$\Phi^m(\xi) = \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})].$$

Je možné ukázať, že ak $\Theta$ je interval, tak minimaxné kritérium sa dá prepísať takto

$$\Phi^{m}(\xi) = \sup_{g \text{ na } \Theta} \int_{\Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] g(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

kde suprémum je cez všetky hustoty g(.) na  $\Theta$ . Suprémum sa nedosahuje pre žiadnu hustotu g(.) (teoreticky by mala zodpovedať tzv. Diracovej hustote). Vhodnou aproximáciou môžeme dosiahnuť, že suprémum sa zmení na maximum. Jednou z možností, ako vhodne zvoliť hustotu  $g(\boldsymbol{\theta})$ , je entropijná regularizácia. Entropia má tvar

$$-\int_{\Theta}g(\boldsymbol{\theta})ln[g(\boldsymbol{\theta})]d\boldsymbol{\theta}$$

Pomocou entropie môžeme minimaxné kritérium aproximovať nasledovne

$$\Phi^{m}(\xi) \doteq \max_{g \text{ na } \Theta} \left\{ \int_{\Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] g(\boldsymbol{\theta}) d\boldsymbol{\theta} - \frac{1}{\lambda} \int_{\Theta} g(\boldsymbol{\theta}) ln[g(\boldsymbol{\theta})] d\boldsymbol{\theta} \right\},$$

kde  $\lambda > 0$  je dané <br/>a $\int_{\Theta} g(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1$ . Vidíme, že sme minimaxné kritérium previedli na úlohu s viazaným extrémom, ktorú vyriešime Lagrangeovou metódou.

Lagrangeova funkcia má tvar

$$\max_{g \text{ na } \Theta} \left\{ \int_{\Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] g(\boldsymbol{\theta}) d\boldsymbol{\theta} - \frac{1}{\lambda} \int_{\Theta} g(\boldsymbol{\theta}) ln[g(\boldsymbol{\theta})] d\boldsymbol{\theta} + \mu \left[ \int_{\Theta} g(\boldsymbol{\theta}) d\boldsymbol{\theta} - 1 \right] \right\},$$

kde  $\mu$  je Lagrangeov multiplikátor, ktorý stanovíme dodatočne z podmienky. Hustotu  $g(\boldsymbol{\theta})$  nahradíme výrazom  $g(\boldsymbol{\theta}) + \alpha \delta(\boldsymbol{\theta})$ , kde  $\delta(.)$  je ľubovoľná hustota,  $\alpha \in \mathbb{R}$ . Výraz zderivujeme podľa  $\alpha$  v bode  $\alpha = 0$ . Pre  $\forall_{\delta(.)}$ 

$$\frac{d}{d\alpha} \left\{ \int_{\Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})][g(\boldsymbol{\theta}) + \alpha\delta(\boldsymbol{\theta})]d\boldsymbol{\theta} - \frac{1}{\lambda} \int_{\Theta} [g(\boldsymbol{\theta}) + \alpha\delta(\boldsymbol{\theta})]ln[g(\boldsymbol{\theta}) + \alpha\delta(\boldsymbol{\theta})]d\boldsymbol{\theta} + \mu \left[ \int_{\Theta} [g(\boldsymbol{\theta}) + \alpha\delta(\boldsymbol{\theta})]d\boldsymbol{\theta} - 1 \right] \right\}_{\alpha=0} = 0.$$

Pre  $\forall_{\delta(.)}$  dostávame

$$\int_{\Theta} \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})]\delta(\boldsymbol{\theta})d\boldsymbol{\theta} - \frac{1}{\lambda}\int_{\Theta} \delta(\boldsymbol{\theta})\ln[g(\boldsymbol{\theta})]d\boldsymbol{\theta} - \frac{1}{\lambda}\int_{\Theta} g(\boldsymbol{\theta})\frac{\delta(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}d\boldsymbol{\theta} + \mu\int_{\Theta} \delta(\boldsymbol{\theta})d\boldsymbol{\theta} = 0.$$

Definujeme  $H(\boldsymbol{\theta})$ ,

$$H(\boldsymbol{\theta}) = \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})] - \frac{1}{\lambda} ln[g(\boldsymbol{\theta})] - \frac{1}{\lambda} + \mu,$$

predošlú rovnosť môžeme písať v tvare

$$\int_{\Theta} H(\boldsymbol{\theta}) \delta(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0.$$

Keďže výraz platí pre  $\forall_{\delta(.)}$ , tak z toho vyplýva, že  $H(\boldsymbol{\theta}) = 0$ , čo vedie k nasledujúcim rovnostiam

$$H(\boldsymbol{\theta}) = \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})] - \frac{1}{\lambda} ln[g^*(\boldsymbol{\theta})] - \frac{1}{\lambda} + \mu = 0,$$
$$ln[g^*(\boldsymbol{\theta})] = \lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})] - \frac{\lambda}{\lambda} + \lambda\mu,$$
$$g^*(\boldsymbol{\theta}) = e^{\lambda\mu - 1} e^{\lambda\Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]}.$$

Konštantu $e^{\lambda\mu-1}$ vyrátame z podmienky  $\int_{\Theta}g^{*}(\pmb{\theta})d\pmb{\theta}=1$ 

$$\int_{\Theta} g^*(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1,$$
$$\int_{\Theta} e^{\lambda \mu - 1} e^{\lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]} d\boldsymbol{\theta} = 1,$$
$$e^{\lambda \mu - 1} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]} d\boldsymbol{\theta} = 1,$$
$$e^{\lambda \mu - 1} = \frac{1}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]} d\boldsymbol{\theta}}.$$

Po týchto úpravách sme našli hustotu  $g^*(\boldsymbol{\theta})$ , ktorá maximalizuje náš výraz a jej tvar je nasledovný

$$g^*(\boldsymbol{\theta}) = \frac{e^{\lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]} d\boldsymbol{\theta}}.$$

Po nájdení hustoty, ktorá nám maximalizuje naše modifikované minimaxné kritérium ho nasledujúcimi úpravami upravíme do finálnej podoby a teda vytvoríme nové kritérium, ktoré budeme označovať  $\Phi^e(\xi)$ .

$$\Phi^{m}(\xi) \doteq \int_{\Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] g^{*}(\boldsymbol{\theta}) d\boldsymbol{\theta} - \frac{1}{\lambda} \int_{\Theta} g^{*}(\boldsymbol{\theta}) ln[g^{*}(\boldsymbol{\theta})] d\boldsymbol{\theta},$$

kde  $g^*$  už nie je nejaká hustota, ale je to nami vyrátaná hustota  $g^*(\boldsymbol{\theta}) = \frac{e^{\lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\boldsymbol{\xi}, \boldsymbol{\theta})]} d\boldsymbol{\theta}}.$ 

$$\begin{split} \Phi^{m}(\xi) \doteq \frac{\int_{\Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}} - \frac{\frac{1}{\lambda} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} [\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] - \ln[\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}]] d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}} - \frac{\frac{1}{\lambda} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}} - \frac{\lambda}{\lambda} \frac{\int_{\Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}} + \frac{1}{\lambda} \frac{\int_{\Theta} \ln[\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta}}. \end{split}$$

Na záver teda predostrieme finálnu aproximáciu minimaxného kritéria

$$\Phi^*(\xi) \doteq \Phi^e(\xi) = \frac{1}{\lambda} ln \left[ \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta} \right]$$

Minimaxný optimálny návr<br/>h $\xi^m = \arg\min_{\xi\in\Xi} \Phi^m(\xi)$ teda aproximujeme výrazom

$$\xi^m \doteq \xi^e = \arg\min_{\xi \in \Xi} \Phi^e(\xi),$$

teda

$$\xi^{e} = \arg\min_{\xi\in\Xi} \frac{1}{\lambda} ln \left[ \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})]} d\boldsymbol{\theta} \right]$$

čo môžeme označiť ako entropijný optimálny návrh.

Smerovú deriváciu funkcie  $\Phi^e$ , za predpokladu existencie gradientu, uvádzame v nasledujúcej vete, ku ktorej prikladáme aj dôkaz. **Veta 3.7.** Ak existuje gradient  $\nabla \Phi[\mathbf{M}(\mu, \boldsymbol{\theta})]$  pre všetky  $\boldsymbol{\theta} \in \Theta$ , tak potom smerová derivácia funkcie  $\Phi^e$  v bode  $\mu$  a v smere  $\xi$  je

$$\partial \Phi^{e}(\mu,\xi) = \frac{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]} \{ tr\{\nabla \Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})] \boldsymbol{M}(\xi,\boldsymbol{\theta})\} - c(\mu,\boldsymbol{\theta}) \} d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]} d\boldsymbol{\theta}}$$

 $kde \ tr\{\nabla\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]\boldsymbol{M}(\xi,\boldsymbol{\theta})\} = \sum_{x\in\mathscr{X}}\xi(x)\boldsymbol{f}^{\top}(x,\boldsymbol{\theta})\nabla\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]\boldsymbol{f}(x,\boldsymbol{\theta}),$  $c(\mu,\boldsymbol{\theta}) \equiv tr\{\nabla\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]\boldsymbol{M}(\mu,\boldsymbol{\theta})\} = \sum_{x\in\mathscr{X}}\mu(x)\boldsymbol{f}^{\top}(x,\boldsymbol{\theta})\nabla\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]\boldsymbol{f}(x,\boldsymbol{\theta}).$ 

Dôkaz:

$$\begin{split} \partial \Phi^{e}(\mu,\xi) &= \frac{d\Phi^{e}[(1-\beta)\mu+\beta\xi]}{d\beta} \bigg|_{\beta=0} \\ &= \frac{d\frac{1}{\lambda}ln\left[\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}[(1-\beta)\mu+\beta\xi,\boldsymbol{\theta}]]}d\boldsymbol{\theta}\right]}{d\beta} \bigg|_{\beta=0} \\ &= \frac{1}{\lambda\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}[(1-\beta)\mu+\beta\xi,\boldsymbol{\theta}]]}d\boldsymbol{\theta}} \frac{d\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}[(1-\beta)\mu+\beta\xi,\boldsymbol{\theta}]]}d\boldsymbol{\theta}}{d\beta} \bigg|_{\beta=0} \\ &= \frac{\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]}\partial\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta}),\boldsymbol{M}(\xi,\boldsymbol{\theta})]d\boldsymbol{\theta}}{\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]}d\boldsymbol{\theta}} \\ &= \frac{\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]}tr\{\nabla\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})][\boldsymbol{M}(\xi,\boldsymbol{\theta})-\boldsymbol{M}(\mu,\boldsymbol{\theta})]\}d\boldsymbol{\theta}}{\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]}d\boldsymbol{\theta}} \\ &= \frac{\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]}tr\{\nabla\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})][\boldsymbol{M}(\xi,\boldsymbol{\theta})-\boldsymbol{M}(\mu,\boldsymbol{\theta})]\}d\boldsymbol{\theta}}{\int_{\Theta}e^{\lambda\Phi[\boldsymbol{M}(\mu,\boldsymbol{\theta})]}d\boldsymbol{\theta}}, \end{split}$$

čo je ekvivalentné s tvrdením vety.

#### Nutné a postačujúce podmienky optimality

Návrh  $\xi^e$  je  $\Phi^e$ -optimálny (entropijný optimálny návrh) práve vtedy, keď

$$\Phi^e(\xi^e) = \min_{\xi \in \Xi} \Phi^e(\xi).$$

Veta 3.8. Návrh  $\xi^e$  je  $\Phi^e$ -optimálny práve vtedy, keď

$$\forall_{\xi} \quad \partial \Phi^e(\xi^e, \xi) \ge 0.$$

Skúsme si Vetu 3.8 podrobne rozpísať a odvodiť konkrétnejšie pravidlo optimality.

$$\begin{aligned} \forall_{\xi} \quad \partial \Phi^{e}(\xi^{e},\xi) \geq 0 \Leftrightarrow \forall_{\xi} \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{tr\{\nabla \Phi[M(\xi^{e},\theta)]M(\xi,\theta)\} - c(\xi^{e},\theta)\} d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \geq 0 \\ \Leftrightarrow \forall_{\xi} \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} tr\{\nabla \Phi[M(\xi^{e},\theta)]M(\xi,\theta)\} d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \geq 0 \\ \geq \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{c(\xi^{e},\theta)\} d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \\ \Leftrightarrow \forall_{\xi} \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \sum_{x \in \mathscr{X}} \xi(x) f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta) d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \geq 0 \\ \geq \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \sum_{x \in \mathscr{X}} \xi^{e}(x) f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta) d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \\ \Leftrightarrow \forall_{x \in \mathscr{X}} \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta)\} d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \\ \geq \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \sum_{x \in \mathscr{X}} \xi^{e}(x) f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta)] d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \\ \geq \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta)\} d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \\ = \quad \frac{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta)\} d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \\ \Leftrightarrow \quad \frac{\min_{x \in \mathscr{X}} \int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta)\} d\theta}{\int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} d\theta} \\ \Leftrightarrow \quad \min_{x \in \mathscr{X}} \int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{f^{\top}(x,\theta) \nabla \Phi[M(\xi^{e},\theta)]f(x,\theta)\} d\theta} \\ = \quad = \quad \int_{\Theta} e^{\lambda \Phi[M(\xi^{e},\theta)]} \{c(\xi^{e},\theta)\} d\theta, \end{aligned}$$

a teda platí

Veta 3.9. Nech existuje gradient  $\nabla \Phi[\mathbf{M}(\xi^e, \boldsymbol{\theta})]$  pre všetky  $\boldsymbol{\theta} \in \Theta$ . Potom návrh  $\xi^e$  je  $\Phi^e$ -optimálny práve vtedy, keď

$$\min_{x \in \mathscr{X}} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi^{e}, \boldsymbol{\theta})]} \{ \boldsymbol{f}^{\top}(x, \boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi^{e}, \boldsymbol{\theta})] \boldsymbol{f}(x, \boldsymbol{\theta}) \} d\boldsymbol{\theta} = \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi^{e}, \boldsymbol{\theta})]} \{ c(\xi^{e}, \boldsymbol{\theta}) \} d\boldsymbol{\theta},$$
  
kde  $c(\xi^{e}, \boldsymbol{\theta}) = \sum_{x \in \mathscr{X}} \xi^{e}(x) \boldsymbol{f}^{\top}(x, \boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi^{e}, \boldsymbol{\theta})] \boldsymbol{f}(x, \boldsymbol{\theta}).$ 

Pri samotnom konštruovaní (približne) entropijného optimálneho návrhu budeme postupovať podobne ako pri hľadaní (približne) AVE optimálneho návrhu v Kapitole 2.

Narozdiel od algoritmu uvedeného v Kapitole 2 ako zastavovacie pravidlo budeme používať nasledujúci výraz

$$\left| \frac{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_{n},\boldsymbol{\theta})]} \{c(\xi_{n},\boldsymbol{\theta})\} d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_{n},\boldsymbol{\theta})]} d\boldsymbol{\theta}} - \frac{\min_{x \in \mathscr{X}} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_{n},\boldsymbol{\theta})]} \{\boldsymbol{f}^{\top}(x,\boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi_{n},\boldsymbol{\theta})] \boldsymbol{f}(x,\boldsymbol{\theta})\} d\boldsymbol{\theta}}{\int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_{n},\boldsymbol{\theta})]} d\boldsymbol{\theta}} \right| \leq \epsilon_{stop}, \quad (14)$$

kde  $\{c(\xi_n, \boldsymbol{\theta})\} = \sum_{x \in \mathscr{X}} \xi_n(x) \boldsymbol{f}^{\top}(x, \boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi_n, \boldsymbol{\theta})] \boldsymbol{f}(x, \boldsymbol{\theta})$ . Keď náš návrh nespĺňa pravidlo zastavenia, potrebujeme do návrhu pridať bod  $x_{min}$ , ktorý je definovaný nasledovne

$$x_{min} = \arg\min_{x\in\mathscr{X}} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_n,\boldsymbol{\theta})]} \{ \boldsymbol{f}^{\top}(x,\boldsymbol{\theta}) \nabla \Phi[\boldsymbol{M}(\xi_n,\boldsymbol{\theta})] \boldsymbol{f}(x,\boldsymbol{\theta}) \} d\boldsymbol{\theta}.$$

### 4 Riešenie príkladov

#### 4.1 Priklad 1

#### Michaelis-Mentenov model rýchlosti enzymatickej reakcie

Uvažujme jednoparametrický Michaelis-Mentenov model, ktorý opisuje rýchlosť enzymatickej reakcie [6]. Model má tvar

$$y_x = \frac{x}{\theta + x} + \varepsilon_x.$$

Teda

$$\eta(x,\theta) = \frac{x}{\theta + x},$$
$$\boldsymbol{f}(x,\theta) = \frac{\partial \eta(x,\theta)}{\partial \theta} = -\frac{x}{(\theta + x)^2}.$$

Informačná matica má tvar

$$M(\xi, \theta) = \sum_{x \in \mathscr{X}} \xi(x) \frac{x^2}{(\theta + x)^4}.$$

Keďže je tento príklad jednoparametický, tak informačná matica je iba skalár. V tomto príklade budeme ako lokálnu kritériálnu funkciu používať asymptotickú varianciu, teda

 $\Phi[\boldsymbol{M}(\xi,\theta)] = \boldsymbol{M^{-1}}(\xi,\theta).$ 

Máme dané

$$\Theta = \{\theta; 1 \le \theta \le 2\},$$
$$\mathscr{X} = \{x; 1 \le x \le 2\}.$$

Kritéria, ktoré používame

$$B(\xi, \pi) = E_{\pi} \{ \Phi[\boldsymbol{M}(\xi, \theta)] \},$$
$$\Phi^{m}(\xi) = \max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi, \theta)],$$
$$\Phi^{e}(\xi) = \frac{1}{\lambda} ln \left[ \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \theta)]} d\boldsymbol{\theta} \right].$$

#### 4.1.1 Riešenie pomocou H-algoritmu [6]

Pre skonštruovanie (približne) minimaxného optimálneho návrhu potrebujeme zadefinovať pomocnú sieť bodov  $h_1 = 0.25, h_2 = 0.5, h_3 = 0.75$  a  $h_4 = 1$ . Takisto zvolíme počiatočné apriórne rozdelenie

$$\pi^{(1)} = 1$$
 pre  $\theta = 1$ .

Pre jednoduchosť budeme písať  $\pi^{(1)}(\theta)$ :  $\pi^{(1)}(1) = 1$ . Je potrebné zvoliť aj  $\epsilon_{stop}$ , teda maximálnu "vzdialenosť" od AVE optimálneho návrhu, ktorú pri hľadaní približne AVE optimálneho návrhu akceptujeme. Pokúsili sme sa hľadať AVE optimálne návrhy, teda  $\epsilon_{stop}$  sme zvolili rovné nule. AVE optimálny návrh prislúchajúci apriórnemu rozdeleniu  $\pi^{(1)}$  je jednobodový návrh  $\xi^{\pi^{(1)}}(1) = 1$ , resp.

$$\xi^{\pi^{(1)}} = \begin{cases} 1\\ 1 \end{cases}.$$

Overíme pravidlo zastavenia (6), teda vyčíslime  $B(\pi^{(1)}) = 16$  a nájdeme

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(1)}}, \theta)] = \max_{\theta \in \Theta} (\theta + 1)^4.$$

Predchádzajúci výraz nadobúda maximum v bod<br/>e $\theta\doteq2$ a tým maximom je hodnota 80.9893. Vidíme, že

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(1)}}, \theta)] = 80.9893 \nleq B(\pi^{(1)}) = 16,$$

a teda pravidlo zastavenia (6) nie je splnené pre všetky  $\theta \in \Theta$  (našli sme jedno také) a preto pokračujeme v algoritme ďalej. Pomocou vyššie definovanej pomocnej siete vygenerujeme 4 nové apriórne rozdelenia, ktoré môžeme vidieť v Tabuľke 1. AVE optimálne návrhy prislúchajúce každému rozdeleniu sú jednobodové a spolu s vyčíslenými hodnotami  $B(\pi_t^{(2)})$  taktiež uvedené v Tabuľke 1. Keďže všetky návrhy sú jednobodové, tak v tabuľke uvádzame len body x, kde bod  $x^{\pi_t^{(l)}}$  je bod v návrhu  $\xi^{\pi_t^{(l)}}$ . Chceme podotknúť, že pri riešení príkladov niekedy uvádzame aj hodnotu  $\epsilon^{\pi_t^{(l)}}$  (resp.  $\epsilon^{\pi^{(l)}}$ ). Táto hodnota reprezentuje "vzdialenosť" konkrétneho návrhu  $\xi^{\pi_t^{(l)}}$  (resp.  $\xi^{\pi^{(l)}}$ ) od optimálneho návrhu. Ako môžeme vidieť, výraz  $B(\pi_t^{(2)})$  je najväčší pre t=4 a tak označíme  $\pi_4^{(2)} = \pi^{(2)}$ . Návrh  $\xi^{\pi^{(2)}}$  definujeme ako  $\xi^{\pi^{(2)}} = \xi^{\pi_4^{(2)}}$ , teda AVE optimálny návrh pre apriórne rozdelenie  $\pi^{(2)}$ 

t	$\pi_t^{(2)}(1)$	$\pi_t^{(2)}(2)$	$x^{\pi_t^{(2)}}$	$B(\pi_t^{(2)})$	$\epsilon^{\pi_t^{(2)}}$
1	0.75	0.25	1.4794	29.6901	0
2	0.5	0.5	1.7190	41.6144	0
3	0.25	0.75	1.8800	52.9549	0
4	0	1.0	1.9999	63.9958	0

Tabuľka 1: Pr. 1, H-algoritmus. Výsledky druhej iterácie

je jednobodový návrh  $\xi^{\pi^{(2)}}(1.9999) = 1$ , resp.

$$\xi^{\pi^{(2)}} = \begin{cases} 1.9999\\ 1 \end{cases}$$

Hodnota AVE kritéria tohto návrhu je  $B(\pi^{(2)}) = 63.9958$ . Krok 4, teda

$$B(\pi^{(2)}) = 63.9958 \ge B(\pi^{(1)}) = 16$$

je splnený a tak máme kandidáta na minimaxný optimálny návrh. Preveríme, či tento kandidát spĺňa pravidlo zastavenia (6). Nájdeme

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(2)}}, \theta)] = \max_{\theta \in \Theta} \frac{(\theta + 2)^4}{4}.$$

Týmto maximom je hodnota<sup>1</sup> 63.995769224423420

$$\max_{\theta \in \Theta} \Phi[\mathbf{M}(\xi^{\pi^{(2)}}, \theta)] = 63.995769224423420 \le B(\pi^{(2)}) = 63.9958.$$

Vidíme, že nerovnosť platí pre maximum, tak je jasné, že táto nerovnosť platí pre všetky  $\theta \in \Theta$  a preto pravidlo zastavenia (6), ktoré má tvar

$$\Phi[\boldsymbol{M}(\xi^{\pi^{(l)}},\boldsymbol{\theta})] \leq B(\pi^{(l)}) \text{ pre všetky } \boldsymbol{\theta} \in \Theta,$$

je pre návrh  $\xi^{\pi^{(2)}}$  splnené. Môžeme konštatovať, že rozdelenie  $\pi^{(2)}$  je "najnepriaznivejšie rozdelenie" a návrh  $\xi^{\pi^{(2)}}$  je minimaxný optimálny návrh, teda  $\xi^m = \xi^{\pi^{(2)}}$ . Hodnota minimaxného kritéria návrhu  $\xi^m$  je  $\Phi^m(\xi^m) = 63.995769224423420$  a čas skonštruovania je 0.018354 sekúnd.

<sup>&</sup>lt;sup>1</sup>Matlab používa numerické metódy pri výpočte maxima. Tieto hodnoty ponechávame v takomto tvare (nezaokrúhlime ich), pretože ich chceme porovnať s hodnotami získanými entropijnou regularizáciou.

4.1

Príklad 1

Tento príklad prezentuje sám autor H-algoritmu H. Nyquist v článku [6]. Výsledky prvej iterácie sa zhodujú s našimi výsledkami. Výsledky druhej iterácie, ktoré autor prezentuje, sa však s našimi už nezhodujú. Po dôkladnom preskúmaní sme dospeli k záveru, že autor Nyquist pri konštruovaní minimaxného optimálneho návrhu niekde pochybil a medzivýsledky druhej iterácie prezentované v článku [6] sú nesprávne (pre t=1, 2, 3), no aj napriek chybným medzivýsledkom autor dospel k rovnakému záveru ako my, teda minimaxný optimálny návrh sa zhoduje s naším minimaxným optimálnym návrhom. V Tabuľke 2 uvádzame Nyquistove výsledky druhej iterácie z článku [6] spolu so správnymi hodnotami AVE kritéria návrhov, ktoré autor Nyquist označil ako AVE optimálne.

	Nyquistove výsledky				správne	výsledky
t	$\pi_t^{(2)}(1)$	$\pi_t^{(2)}(2)$	$x^{\pi_t^{(2)}}$	$B(\pi_t^{(2)})$	$B(\pi_t^{(2)})$	$\epsilon^{\pi_t^{(2)}}$
1	0.75	0.25	1.0435	19.9979	31.7101	2.7252
2	0.5	0.5	1.1271	26.5465	45.6942	5.1772
3	0.25	0.75	1.3355	38.7690	56.2199	3.7015
4	0	1	2.0000	64.000	64.000	0

Tabuľka 2: Pr. 1, H-algoritmus. Výsledky druhej iterácie podľa H. Nyquista

Z Tabuľky 2 môžeme vidieť, že najväčšia hodnota AVE kritéria patrí návrhu, ktorý autor Nyquist skonštruoval pre apriórne rozdelenie  $\pi_4^{(2)}$  a tak  $\pi^{(2)} = \pi_4^{(2)}$  a  $\xi^{\pi^{(2)}} = \xi^{\pi_4^{(2)}}$ . AVE optimálny návrh  $\xi^{\pi^{(2)}}$  má tvar  $\xi^{\pi^{(2)}}(2) = 1$ , resp.

$$\xi^{\pi^{(2)}} = \begin{cases} 2\\ 1 \end{cases}.$$

Hodnota AVE kritéria návrhu  $\xi^{\pi^{(2)}}$  je  $B(\pi^{(2)}) = 64$ . Krok 4 je splnený, pretože platí

$$B(\pi^{(2)}) = 64 \ge B(\pi^{(1)}) = 16.$$

Overíme pravidlo zastavenia (6)

$$\max_{\theta \in \Theta} \Phi[\mathbf{M}(\xi^{\pi^{(2)}}, \theta)] = 64 \le B(\pi^{(2)}) = 64.$$
Keďže nerovnosť platí pre maximum, platí pre všetky  $\theta \in \Theta$  a tak pravidlo zastavenia (6) je splnené. Rozdelenie  $\pi^{(2)}$  je "*najnepriaznivejšie rozdelenie*", návrh  $\xi^{\pi^{(2)}}$  je minimaxný optimálny návrh, teda  $\xi^m = \xi^{\pi^{(2)}}$  a hodnota minimaxného kritéria tohto návrhu je  $\Phi^m(\xi^m) = 64$ .

	opravené Nyquistove výsledky				naše v	ýsledky
t	$\pi_t^{(2)}(1)$	$\pi_t^{(2)}(2)$	$x^{\pi_t^{(2)}}$	$B(\pi_t^{(2)})$	$x^{\pi_t^{(2)}}$	$B(\pi_t^{(2)})$
1	0.75	0.25	1.0435	31.7101	1.4794	29.6901
2	0.5	0.5	1.1271	45.6942	1.7190	41.6144
3	0.25	0.75	1.3355	56.2199	1.8800	52.9549
4	0	1	2.0000	64.000	2.000	64.000

Tabuľka 3: Pr. 1, H-algoritmus. Porovnanie našich a Nyquistových výsledkov druhej iterácie

V Tabuľke 3 porovnávame výsledky, ku ktorým sme sa dopracovali my spolu s opravenými Nyquistovými výsledkami. Z tejto tabuľky je jasné, že pri jednotlivých apriórnych rozdeleniach pre t=1, 2, 3 sme my našli návrhy, ktorých hodnota kritéria je menšia ako hodnota kritéria návrhov, ktoré uvádza autor. Z týchto tabuliek je teda jasné, že návrhy prezentované v článku [6] pre t=1, 2, 3 nie sú AVE optimálne. Tento postreh je zreteľný už v Tabuľkách 1 a 2, kde môžeme vidieť, že "vzdialenosti" nami skonštruovaných návrhov od optimálnych sú nulové, zatiaľčo v autorovom prípade nulové nie sú. Pre t=1, 2, 3 autor Nyquist skonštruoval len približne AVE optimálne návrhy a nie AVE optimálne návrhy tak, ako sme to spravili my. Najdôležitejší je však AVE optimálny návrh pre rozdelenie, ktoré sme nazvali *"najnepriaznivejšie rozdelenie"*. Naše výsledky (po zaokrúhlení) sa pre toto rozdelenie s výsledkami autora zhodujú, a tak môžeme konštatovať, že minimaxný optimálny návrh, ktorý sme skonštruovali, je totožný s minimaxným optimálnym návrhom, ktorý skonštruoval autor článku Nyquist.

#### 4.1.2 Riešenie pomocou entropijnej regularizácie

Pri skonštruovaní (približne) minimaxného optimálneho návrhu pomocou entropijnej regularizácie ((približne) entropijný optimálny návrh) je takisto potrebné zvoliť  $\epsilon_{stop}$ -maximálnu "vzdialenosť" nášho návrhu od optimálneho, ktorú akceptujeme. Aj pri tomto algoritme sme sa pokúsili nájsť nie približne optimálny návrh, ale optimálny návrh, takže opäť sme zvolili  $\epsilon_{stop}$  rovné nule. Ako môžeme vidieť v Podkapitole 3.2, ktorá sa celá venuje odvodeniu vzorca, ktorým aproximujeme minimaxné kritérium, je pri výpočtoch potrebné zvoliť parameter  $\lambda$ . V tejto časti sa pokúsime nájsť entropijný optimálny návrh a analyzovať citlivosť algoritmu na parameter  $\lambda$ . Tento model je jednoparametrický a preto sme sa pri konštrukcii optimálneho návrhu pokúsili body do návrhu nie pridávať, ale priamo ich zamieňať tak, ako to je popísané v nasledujúcom texte. Ako pravdilo zastavenia používame pravidlo (14), ktorého podrobné odvodenie uvádzame v Podkapitole 3.2.

# Podrobná analýza algoritmu pri konštrukcii entropijného optimálneho návrhu pre parameter $\lambda=1$

Začneme jednobodovým návrhom  $\xi_0(1) = 1$ , resp.

$$\xi_0 = \begin{cases} 1 \\ 1 \end{cases}.$$

Overíme pravidlo zastavenia (14), teda vyčíslime "vzdialenosť" návrhu  $\xi_0$  od optimálneho (značíme  $\epsilon_0$ ). Hodnota "vzdialenosti" je  $\epsilon_0 = 20.9334 \leq \epsilon_{stop} = 0$  a vidíme, že tento návrh nie je optimálny a preto bod 1 v našom návrhu zameníme bodom

$$x_{min} = \arg\min_{x\in\mathscr{X}} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_0,\theta)]} \{ \boldsymbol{f}^{\top}(x,\theta) \nabla \Phi[\boldsymbol{M}(\xi_0,\theta)] \boldsymbol{f}(x,\theta) \} d\theta = 1.9906886,$$

teda dostávame

$$\xi_1 = \left\{ \begin{array}{c} 1.9906886\\ 1 \end{array} \right\}.$$

Vyčíslime "vzdialenosť" tohto návrhu, ted<br/>a $\epsilon_1 = 3.3328 * 10^{-4}$ . Opäť vidíme, že $\epsilon_1$ nie je rovné nule a tak pokračujeme v algoritme ďalej. Do návrhu pridáme bod

$$x_{min} = \arg\min_{x \in \mathscr{X}} \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_1, \theta)]} \{ \boldsymbol{f}^{\top}(x, \theta) \nabla \Phi[\boldsymbol{M}(\xi_1, \theta)] \boldsymbol{f}(x, \theta) \} d\theta = 1.9842217,$$
$$\xi_2 = \left\{ \begin{array}{c} 1.9842217\\1 \end{array} \right\}.$$

"Vzdialenosť" návrhu $\xi_2$ od optimálneho je $\epsilon_2=6.1678*10^{-9},$ teda znova nájdeme bod $x_{min}$ a pridáme ho do návrhu

$$\xi_3 = \left\{ \begin{array}{c} 1.9842482\\ 1 \end{array} \right\}.$$

 $\epsilon_3 = 2.2794 * 10^{-12}$  a  $x_{min} = 1.9842480$ .

$$\xi_4 = \left\{ \begin{array}{c} 1.9842480\\ 1 \end{array} \right\} \,.$$

Opäť overíme pravidlo zastavenia (14) pre návrh  $\xi_4$ , čiže vyčíslime  $\epsilon_4 = 0 \leq \epsilon_{stop} = 0$ . Môžeme vidieť, že tento návrh spĺňa pravidlo zastavenia a preto návrh  $\xi_4$  je entropijný optimálny návrh, teda  $\xi^e = \xi_4$ . Hodnota entropijného kritéria optimálneho návrhu je  $\Phi^e(\xi^e) = 59.851122032252$ . Hodnota minimaxného kritéria entropijného optimálneho návrhu je  $\Phi^m(\xi^e) = 63.99775316607$ . Všetky tieto výpočty môžeme vidieť v Tabuľke 4, kde prvý stĺpec t predstavuje číslo iterácie. Keďže všetky návrhy sú jednobodové, podobne ako v predchádzajúcich tabuľkách, uvádzame len body x, kde bod  $x_t$  je bod návrhu  $\xi_t$ . V tabuľke je uvedená aj "vzdialenosť" návrhu  $\xi_t$  od optimálneho, teda  $\epsilon_t$ .

hodnota parametra $\lambda = 1$			
t	$x_t$	$\epsilon_t$	
0	1	20.9334	
1	1.9906886	$3.3328 * 10^{-4}$	
2	1.9842217	$6.1678 * 10^{-9}$	
3	1.9842482	$2.2794 * 10^{-12}$	
4	1.9842480	0	

**Tabuľka 4:** Pr. 1, entropijná regularizácia. Výsledky pre hodnotu parametra  $\lambda = 1$ 

V nasledujúcich tabuľkách uvedieme výpočty pre rôzne hodnoty parametra  $\lambda$ . Musíme podotknúť, že pre hodnotu  $\lambda > 8$  sa pri výpočte integrálu vyskytuje číslo väčšie ako realmax = 1.797693134862316 \* 10<sup>308</sup>. Číslo realmax je v programe Matlab maximálne reálne číslo, teda ak sa vyskytne číslo väčšie ako realmax, program Matlab vníma toto číslo ako nekonečno a preto budeme analyzovať výpočty, kde je parameter  $\lambda$  z intervalu (0,8). V tabuľkách uvádzame číslo iterácie t, bod v návrhu  $\xi_t$ , ktorý značíme  $x_t$ a ako poslednú hodnotu uvádzame "vzdialenosť" návrhu  $\xi_t$  od optimálneho, teda  $\epsilon_t$ . V Tabuľke 7 môžeme vidieť záverečné porovnanie entropijných optimálnych návrhov pre rôzne hodnoty parametra  $\lambda$ , kde navyše uvádzame hodnoty entropijného a minimaxného kritéria. Z nasledujúcich tabuliek nie je možné určiť, či nám náš algoritmus so zvyšujúcou sa hodnotou parametra  $\lambda$  skonverguje rýchlejšie. Je však zrejmé, že čím je hodnota parametra  $\lambda$  vyššia, tým je entropijný optimálny návrh bližšie k minimaxnému optimálnemu návrhu, ktorý sme skonštruovali pomocou H-algoritmu, teda k návrhu  $\xi^m(2) = 1$ . Tieto postrehy môžeme vidieť v grafickej podobe na Obrázku 1.

	hodnota pa	rametra $\lambda = 0.1$	hodnota parametra $\lambda = 2$		
t	$x_t$	$\epsilon_t$	$x_t$	$\epsilon_t$	
0	1	15.6632	1	21.2246	
1	1.9029566	$3.4379 * 10^{-2}$	1.9953561	$8.2088 * 10^{-5}$	
2	1.8353774	$9.9134 * 10^{-5}$	1.9921486	$4.9334 * 10^{-10}$	
3	1.8389386	$2.8401 * 10^{-7}$	1.992154959	$3.3000 * 10^{-13}$	
4	1.8387459	$1.1135 * 10^{-9}$	1.992154946	0	
5	1.8387563	$1.3193 * 10^{-11}$			
6	1.8387557	$8.5760 * 10^{-13}$			
7	1.838755776	$4.3313 * 10^{-14}$			
8	1.838755774	0			

**Tabuľka 5:** Pr. 1, entropijná regularizácia. Výsledky pre hodnotu parametra  $\lambda=0.1,2$ 

	hodnota p	parametra $\lambda = 5$	hodnota parametra $\lambda = 8$		
t	$x_t$	$\epsilon_t$	$x_t$	$\epsilon_t$	
0	1	21.3992	1	21.4429	
1	1.9981443	$1.3002 * 10^{-5}$	1.9988399	$5.0594 * 10^{-6}$	
2	1.9968673	$3.6483 * 10^{-11}$	1.99804277	$1.2514 * 10^{-11}$	
3	1.9968683	0	1.9980431596	$8.9836 * 10^{-15}$	
4			1.9980431594	0	

**Tabuľka 6:** Pr. 1, entropijná regularizácia. Výsledky pre hodnotu parametra  $\lambda = 5, 8$ 



**Obr. 1:** Pr. 1, entropijná regularizácia. Graf závislosti bodu návrhu od parametra  $\lambda$ 

Z Tabuľky 7 môžeme vidieť, že čím je hodnota parametra  $\lambda$  vyššia, tým je hodnota minimaxného kritéria skonštruovaného entropijného optimálneho návrhu nižšia. Takisto vidíme, že so zvyšujúcou hodnotou parametra  $\lambda$  sa hodnota entropijného kritéria zvyšuje a približuje sa k hodnote minimaxného kritéria. Tieto postrehy sú graficky znázornené na Obrázku 2, kde uvádzame graf hodnôt minimaxného a entropijného kritéria entropijného optimálneho návrhu skonštruovaného pri danej hodnote parametra  $\lambda$ . V nasledujúcej tabuľke navyše uvádzame aj čas výpočtu  $\tau$  v sekundách.

$\lambda$	x	$\epsilon$	$\Phi^e$	$\Phi^m$	$ au/ ext{sek.}$
0.1	1.838755774537501	0	46.538718410019364	64.222013225906579	0.994032
1	1.984248047486815	0	59.851122032252285	63.997753166071924	0.541308
2	1.992154945936448	0	61.576455526116298	63.996255211756228	0.562583
5	1.996868329178078	0	62.846728273036717	63.995844521289463	0.538476
8	1.998043159447676	0	63.220362459269190	63.995797850449875	0.653617

**Tabuľka 7:** Pr. 1, entropijná regularizácia. Záverečné porovnanie optimálnych návrhov pre rôzne hodnoty parametra  $\lambda$ 

V Tabuľke 7 si môžeme taktiež všimnúť, že so zvyšujúcou sa hodnotou parametra  $\lambda$  sa hodnota minimaxného a entropijného kritéria približujú k hodnote minimaxného kritéria návrhu, ktorý sme skonštruovali pomocou H-algoritmu, teda k hodnote  $\Phi^m(\xi^m) = 63.995769224423420$ .



**Obr. 2:** Pr. 1, entropijná regularizácia. Graf hodnôt kritérií v závislosti od parametra  $\lambda$ 

#### 4.2 Príklad 2

#### Logistická regresia

Uvažujme model logistickej regresie [6]

$$\eta(x, \boldsymbol{\theta}) = \theta_2(x - \theta_1).$$

Tento príklad má štatistický pôvod, ktorý sa líši od situácie opisovanej v Kapitole 1 a preto si vyžaduje krátky úvodný komentár. Pre merania, ktoré vykonávame v bode x, pozorujeme náhodnú veličinu  $y_x$ , ktorá môže nadobúdať len dve hodnoty: 0 alebo 1. Teda pravdepodobnosť

$$P_x(y_x) = \begin{cases} \Pi(x), & \text{ak } y_x = 1, \\ 1 - \Pi(x), & \text{ak } y_x = 0, \end{cases}$$

kde  $\Pi(x)$  je neznáme. Celý experiment pozostáva z takýchto nezávislých meraní v rôznych bodoch  $x \in \mathscr{X}$  alebo z nezávislých opakovaní meraní v tom istom bode x.

V logistickej regresii sa za regresnú krivku nepovažuje graf<br/> závislosti $\Pi(x)$  od x, ale funkcia

$$x \in \mathscr{X} \mapsto \eta(x) = ln\left(\frac{\Pi(x)}{1 - \Pi(x)}\right),$$

ktorá má svoje zdôvodnenie v teórii binomického rozdelenia pravdepodobnosti. Je zrejmé, že potom

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

Ako môžeme vidieť v zadaní tohto príkladu,  $\eta(x, \theta) = \theta_2(x - \theta_1)$ , kde  $\theta_1$  je z intervalu [0.5, 1.5] a  $\theta_2$  z intervalu [4, 5] sú neznáme parametre.

Vo všeobecných štatistických modeloch závislých od vektora parametrov  $\theta$  je zaužívané určovať kvalitu experimentu pomocou Fischerovej informačnej matice

$$E_{\boldsymbol{\theta}}\left[\frac{\partial ln(P(y|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}}\frac{\partial ln(P(y|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^{\top}}\right],$$

kde  $P(y|\boldsymbol{\theta})$  je pravdepodobnosť pozorovaného vektora y pri danej hodnote parametra  $\boldsymbol{\theta}$ . Špeciálne teda pre uvažovaný experiment v prípade merania v jedinom bode x informačná matica má tvar

$$M(x,\boldsymbol{\theta}) = \Pi(x,\boldsymbol{\theta}) \frac{\partial ln\Pi(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial ln\Pi(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\top}} + (1 - \Pi(x,\boldsymbol{\theta})) \frac{\partial ln(1 - \Pi(x,\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} \frac{\partial ln(1 - \Pi(x,\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^{\top}}$$

Odtiaľ vyplýva, že ak návrh $\xi$ je taký, že  $\xi(x)$  je relatívna početnosť opakovaní meraní v bode x, tak informačná matica návrhu  $\xi$ má tvar

$$M(\xi, \boldsymbol{\theta}) = \sum_{x \in \mathscr{X}} M(x, \boldsymbol{\theta}) \xi(x).$$
(15)

Štandardný výpočet derivácie dáva

$$\frac{\partial ln\Pi(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left[1 - \frac{e^{\eta(x,\boldsymbol{\theta})}}{1 + e^{\eta(x,\boldsymbol{\theta})}}\right] \frac{\partial \eta(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}},$$
$$\frac{\partial ln(1 - \Pi(x,\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} = \left[-\frac{e^{\eta(x,\boldsymbol{\theta})}}{1 + e^{\eta(x,\boldsymbol{\theta})}}\right] \frac{\partial \eta(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}.$$

Po dosadení do (15) dostávame

$$M(\xi, \boldsymbol{\theta}) = \sum_{x \in \mathscr{X}} \xi(x) e^{\eta(x, \boldsymbol{\theta})} \frac{1 + e^{\eta(x, \boldsymbol{\theta})}}{(1 + e^{\eta(x, \boldsymbol{\theta})})^3} \frac{\partial \eta(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial \eta(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\top}},$$
$$M(\xi, \boldsymbol{\theta}) = \sum_{x \in \mathscr{X}} \xi(x) \frac{e^{\eta(x, \boldsymbol{\theta})}}{(1 + e^{\eta(x, \boldsymbol{\theta})})^2} \frac{\partial \eta(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial \eta(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\top}}.$$

Ak označíme  $w(x, \theta) = \frac{e^{\eta(x, \theta)}}{(1+e^{\eta(x, \theta)})^2}$  dostávame sa do situácie z Kapitoly 1, pričom

$$f(x, \theta) = \sqrt{w(x, \theta)} \frac{\partial \eta(x, \theta)}{\partial \theta}$$

V našom príklade teda máme

$$\eta(x, \boldsymbol{\theta}) = \theta_2(x - \theta_1),$$
$$w(x, \boldsymbol{\theta}) = \frac{e^{\theta_2(x - \theta_1)}}{(1 + e^{\theta_2(x - \theta_1)})^2},$$
$$f(x, \boldsymbol{\theta}) = \sqrt{\frac{e^{\theta_2(x - \theta_1)}}{(1 + e^{\theta_2(x - \theta_1)})^2}} \begin{bmatrix} -\theta_2\\ x - \theta_1 \end{bmatrix}$$

a informačná matica má tvar

$$M(\xi, \theta) = \sum_{x \in \mathscr{X}} \xi(x) \frac{e^{\theta_2(x-\theta_1)}}{(1+e^{\theta_2(x-\theta_1)})^2} \begin{bmatrix} \theta_2^2 & -\theta_2(x-\theta_1) \\ -\theta_2(x-\theta_1) & (x-\theta_1)^2 \end{bmatrix}.$$

\_

Ako lokálnu kriteriálnu funkciu použijeme D-optimalitu, teda

$$\Phi[\boldsymbol{M}(\xi,\boldsymbol{\theta})] = -ln[det\boldsymbol{M}(\xi,\boldsymbol{\theta})].$$

Máme dané

$$\Theta = \{ (\theta_1, \theta_2); 0.5 \le \theta_1 \le 1.5, 4 \le \theta_2 \le 5 \},$$
$$\mathscr{X} = \{ x; 0 \le x \le 2 \}.$$

Kritéria, ktoré používame

$$B(\xi, \pi) = E_{\pi} \{ \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})] \},$$
$$\Phi^{m}(\xi) = \max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})],$$
$$\Phi^{e}(\xi) = \frac{1}{\lambda} ln \left[ \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi, \boldsymbol{\theta})]} d\boldsymbol{\theta} \right].$$

#### 4.2.1 Riešenie pomocou H-algoritmu [6]

Príklad 2 (logistická regresia) rieši sám autor H-algoritmu Nyquist v článku [6]. Po dôkladnej kontrole v nasledujúcom texte predostrieme podrobnejšiu analýzu výsledkov z [6].

Definujeme pomocnú sieť bodov  $h_1 = 0.2, h_2 = 0.4, h_3 = 0.6, h_4 = 0.8$  a  $h_5 = 1$ . Zvolíme počiatočné apriórne rozdelenie  $\pi^{(1)}$ 

$$\pi^{(1)} = \begin{cases} 0.25, & \text{pre } (\theta_1, \theta_2) = (0.5, 4), \\ 0.25, & \text{pre } (\theta_1, \theta_2) = (0.5, 5), \\ 0.25, & \text{pre } (\theta_1, \theta_2) = (1.5, 4), \\ 0.25, & \text{pre } (\theta_1, \theta_2) = (1.5, 5). \end{cases}$$

Približne AVE optimálny návrh (<br/>  $\epsilon^{\pi^{(1)}}=0.0002)$ prislúchajúci počiatočnému apriórnemu rozdeleniu je návrh

$$\xi^{\pi^{(1)}} = \left\{ \begin{array}{ccc} 0.26 & 1.00 & 1.74 \\ 0.275 & 0.450 & 0.275 \end{array} \right\}.$$

Overíme pravidlo zastavenia (6), teda vyčíslime hodnotu AVE kritéria  $B(\pi^{(1)}) = 3.6906$ a nájdeme

$$\max_{\boldsymbol{\theta}\in\Theta} \Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(1)}},\boldsymbol{\theta})] = 3.8059.$$

Toto maximum sa nadobúda v bode  $\boldsymbol{\theta}^{(1)} = (\theta_1, \theta_2) = (1, 5)$ . Vidíme, že

$$\max_{\theta \in \Theta} \Phi[\mathbf{M}(\xi^{\pi^{(1)}}, \boldsymbol{\theta})] = 3.8059 \nleq B(\pi^{(1)}) = 3.6906,$$

a teda pravidlo zastavenia (6) nie je splnené pre všetky  $\boldsymbol{\theta} \in \Theta$  (našli sme jedno také) a preto pokračujeme v algoritme ďalej. Vygenerujeme 5 nových apriórnych rozdelení pomocou pomocnej siete bodov tak, ako to je opísané v Kroku 3 Podkapitoly 3.1. Maximálna hodnota AVE kritéria patrí návrhu, ktorý je skonštruovaný pri apriórnom rozdelení pre t=1, teda  $\pi^{(2)} = \pi_1^{(2)}$ .

$$\pi^{(2)} = \begin{cases} 0.2, & \text{pre } (\theta_1, \theta_2) = (0.5, 4), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (0.5, 5), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (1.5, 4), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (1.5, 5), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (1, 5). \end{cases}$$

Približne AVE optimálny návrh (<br/>  $\epsilon^{\pi^{(2)}}=0.0017)$ prislúchajúci apriórnemu rozdeleni<br/>u $\pi^{(2)}$ je návrh

$$\xi^{\pi^{(2)}} = \begin{cases} 0.31 & 0.90 & 1.10 & 1.69\\ 0.272 & 0.228 & 0.228 & 0.272 \end{cases}.$$

Hodnota samotného kritéria návrhu  $\xi^{\pi^{(2)}}$  je  $B(\pi^{(2)}) = 3.6948$ . Je zrejmé, že

$$B(\pi^{(2)}) = 3.6948 \ge B(\pi^{(1)}) = 3.6906,$$

takže Krok číslo 4 v Podkapitole 3.1 je splnený a tak môžeme pokračovať v algoritme. Overíme pravidlo zastavenia (6) a preto vyčíslime

$$\max_{\boldsymbol{\theta}\in\Theta} \Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(2)}},\boldsymbol{\theta})] = 3.7900,$$

ktoré sa nadobúda v bode  $\boldsymbol{\theta}^{(2)} = (\theta_1, \theta_2) = (0.5, 5)$  a zároveň v bode  $\boldsymbol{\theta}^{(2')} = (\theta_1, \theta_2) = (1.5, 5).$ 

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(2)}}, \boldsymbol{\theta})] = 3.7900 \nleq B(\pi^{(2)}) = 3.6948.$$

Pravidlo zastavenia (6) nie je splnené pre všetky  $\boldsymbol{\theta} \in \Theta$ , teda návrh  $\xi^{\pi^{(2)}}$  nie je približne minimaxný optimálny návrh, preto musíme v algoritme pokračovať ďalej nasledujúcim krokom. Generujeme 5 nových apriórnych rozdelení nasledovne

$$\pi_t^{(3)} = (1 - h_t)\pi^{(2)} + h_t(\delta^{(2)} + \delta^{(2')})/2,$$

kde  $\delta^{(2')}$  je Diracova miera v bode  $\theta^{(2')}$ . Apriórne rozdelenie, pre ktoré je hodnota AVE kritéria najväčšia je pre t=5, teda  $\pi^{(3)} = \pi_5^{(3)}$ .

$$\pi^{(3)} = \begin{cases} 0.5, & \text{pre } (\theta_1, \theta_2) = (0.5, 5), \\ 0.5, & \text{pre } (\theta_1, \theta_2) = (1.5, 5). \end{cases}$$

Približne AVE optimálny návrh ( $\epsilon^{\pi^{(3)}} = 0.0005$ ) prislúchajúci apriórnemu rozdeleniu  $\pi^{(3)}$  je návrh

$$\xi^{\pi^{(3)}} = \begin{cases} 0.287 & 1.00 & 1.713 \\ 0.2675 & 0.4650 & 0.2675 \end{cases}$$

Hodnota AVE kritéria tohto návrhu je  $B(\pi^{(3)}) = 3.7728$ . Opäť vidíme, že sa hodnota AVE kritéria zvýšila, teda Krok číslo 4 v Podkapitole 3.1 je opäť splnený a tak môžeme skontrolovať, či návrh  $\xi^{\pi^{(3)}}$  spĺňa pravidlo zastavenia. Predtým je nutné vyčísliť

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(3)}}, \boldsymbol{\theta})] = 3.7728.$$

Hodnota sa nadobúda v bode  $\boldsymbol{\theta}^{(3)} = (\theta_1, \theta_2) = (0.5, 5)$  a zároveň v bode  $\boldsymbol{\theta}^{(3')} = (\theta_1, \theta_2) =$ = (1.5, 5). Je zrejmé, že

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(3)}}, \boldsymbol{\theta})] = 3.7728 \le B(\pi^{(3)}) = 3.7728.$$

Keďže táto nerovnosť platí pre maximum, určite platí pre všetky  $\boldsymbol{\theta} \in \Theta$ . Pravidlo zastavenia (6), teda

$$\Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(3)}},\boldsymbol{\theta})] \leq B(\boldsymbol{\xi}^{\pi^{(3)}}) \text{ pre všetky } \boldsymbol{\theta} \in \Theta,$$

je splnené a to znamená, že rozdelenie  $\pi^{(3)}$  je *"najnepriaznivejšie rozdelenie"* a návrh  $\xi^{\pi^{(3)}}$  je približne minimaxný optimálny návrh, teda  $\xi^m = \xi^{\pi^{(3)}}$ . Hodnota minimaxného kritéria tohto návrhu je  $\Phi^m(\xi^m) = 3.7728$ .

Chceme podotknúť, že o samotný výpočet tohto príkladu pomocou H-algoritmu sme sa pokúšali aj my. Problém, ktorý sa pri skonštruovaní samotného algoritmu vyskytol, bolo hľadanie približne AVE optimálnych návrhov. Autor Nyquist sa v článku [6] vôbec nezmieňuje akým spôsobom konštruoval približne AVE optimálne návrhy a preto sme postup na konštrukciu takéhoto návrhu museli navrhnúť my. Algoritmus na nájdenie približne AVE optimálneho návrhu môžeme vidieť v Kapitole 2. Ako sme v tejto kapitole spomenuli, každý príklad je špecifický a preto je často potrebné hľadať konkrétne reštrikcie, ktoré vedú k nájdeniu približne optimálneho návrhu. V tomto príklade sme najlepšie výsledky dosahovali vtedy, keď sme body do návrhu pridávali symetricky, teda

$$\xi_{n+1} = (1 - \alpha_n) + \frac{\alpha_n}{2} \delta_{x_{min1}}(x) + \frac{\alpha_n}{2} \delta_{x_{min2}}(x).$$

Body  $x_{min1}$  a  $x_{min2}$  určíme nasledovne. Nájdeme bod  $x_{min}$  tak, ako to je popísané v Kroku 2 Kapitoly 2. Určíme stred množiny  $\mathscr{X}$  (značíme  $\mathscr{X}_{stred}$ ) a vzdialenosť v, kde  $v = |\mathscr{X}_{stred} - x_{min}|$ . Body  $x_{min1}$  a  $x_{min2}$  definujeme takto

$$x_{min1} = \mathscr{X}_{stred} - v, \qquad \qquad x_{min2} = \mathscr{X}_{stred} + v$$

V Kroku 2 Kapitoly 2 je uvedené, že zvolíme vhodnú dĺžku kroku  $\alpha_n$ . Štandardne sa dĺžka kroku volí  $\frac{1}{q+1}$  (resp.  $\frac{1}{q+2}$  v našom prípade), kde q je počet bodov návrhu  $\xi_n$ . Keďže sa nám pri týchto krokoch nedarilo dosiahnuť uspokojivé výsledky, tak sme skúsili pracovať s "optimálnym" krokom  $\alpha_n$ , teda s krokom, ktorý minimalizuje hodnotu kritéria, čiže

$$\alpha_n = \arg\min_{\alpha \in (0,1)} B(\xi_{n+1}, \pi) = \arg\min_{\alpha \in (0,1)} \int_{\Theta} \Phi[\boldsymbol{M}(\xi_{n+1}, \boldsymbol{\theta})] d\pi(\boldsymbol{\theta})$$

respektíve

$$\alpha_n = \arg\min_{\alpha \in (0,1)} \int_{\Theta} \Phi[\boldsymbol{M}((1-\alpha)\xi_n + \frac{\alpha_n}{2}\delta_{x_{min1}}(x) + \frac{\alpha_n}{2}\delta_{x_{min2}}(x), \boldsymbol{\theta})]d\pi(\boldsymbol{\theta}).$$

Najlepšie výsledky sme pri konštruovaní približne AVE optimálnych návrhov dosahovali vtedy, keď sme Krok 3 popísaný v Kapitole 2 realizovali v každej iterácii a zároveň sme nerealizovali Krok 4, ktorý hovorí o vymazávaní bodov s "malou" váhou. Tento krok sme nerealizovali, pretože body, ktoré sme pridávali do návrhu, tvorili celkom jasné zhluky. Ak sme v Kroku 2 do návrhu pridali bod s "malou" váhou, tak v Kroku 3 bol tento bod spojený s bodom, ktorý mal váhu dostatočne "veľkú" na to, aby sme ho z návrhu nevymazali. V nasledujúcom texte prezentujeme výsledky, ku ktorým sme sa dopracovali a kvôli prehľadnosti porovnáme "tvárov v tvár" naše výsledky s výsledkami autora Nyquista. Výsledky prezentujeme v tabuľkách, kde uvádzame

- $\xi^{\pi^{(l)}}$  návrh vzhľadom na apriórne rozdelenie  $\pi^{(l)}$ ,
- $\epsilon^{\pi^{(l)}}$ "vzdialenosť" návrhu  $\xi^{\pi^{(l)}}$ od optimálneho návrhu,
- $\Delta_{\min}^{\pi^{(l)}}$  minimálna vzájomná vzdialenosť 2 bodov (popisané v Kapitole 2),
- $B(\pi^{(l)})$  hodnota AVE kritéria návrhu  $\xi^{\pi^{(l)}}$ ,
- $\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(l)}}, \boldsymbol{\theta})]$  hodnota potrebná na overenie pravidla zastavenia (6),

- $\boldsymbol{\theta}^{(l)}$  bod, v ktorom sa nadobúda,  $\max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(l)}}, \boldsymbol{\theta})],$
- $pi^{\pi^{(l)}}$  počet iterácii pri konštrukcii návrhu  $\xi^{\pi^{(l)}}$ ,
- $\tau^{\pi^{(l)}}$ čas výpočtu pri konštrukcii návrhu <br/>  $\xi^{\pi^{(l)}}$ uvedený v sekundách.

Definujeme pomocnú sieť bodov  $h_1 = 0.2, h_2 = 0.4, h_3 = 0.6, h_4 = 0.8$  a  $h_5 = 1$ . Pre počiatočné apriórne rozdelenie  $\pi^{(1)}$ , ktoré má tvar

$$\pi^{(1)} = \begin{cases} 0.25, & \text{pre } (\theta_1, \theta_2) = (0.5, 4), \\ 0.25, & \text{pre } (\theta_1, \theta_2) = (0.5, 5), \\ 0.25, & \text{pre } (\theta_1, \theta_2) = (1.5, 4), \\ 0.25, & \text{pre } (\theta_1, \theta_2) = (1.5, 5), \end{cases}$$

sme skonštruovali návrh, ktorý spolu s ostatnými hodnotami uvádzame v nasledujúcej tabuľke.

	naše výsledky	výsledky autora Nyquista
$\epsilon \pi^{(1)}$	$\int 0.259  1.000  1.741$	$\int 0.26  1.00  1.74$
ς	$0.2749  0.4502  0.2749 \int$	$0.275  0.450  0.275 \int$
$\epsilon^{\pi^{(1)}}$	0.000035	0.0002
$\Delta_{min}^{\pi^{(1)}}$	0.4	
$B(\pi^{(1)})$	3.6906	3.6906
$\max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(1)}}, \boldsymbol{\theta})]$	3.8076	3.8059
$oldsymbol{ heta}^{(1)}=( heta_1, heta_2)$	(1,5)	(1,5)
$pi^{\pi^{(1)}}$	382	
$ au^{\pi^{(1)}}/ ext{sek.}$	24	

Tabuľka 8: Pr. 2, H-algoritmus. Porovnanie výsledkov pre rozdelenie  $\pi^{(1)}$ 

Z Tabuľky 8 môžeme vidieť, že ak by sme hodnoty v našom návrhu zaokrúhlili na toľko desatinných miest ako autor Nyquist, tak by bol náš návrh totožný s autorovým. Naše výsledky chceme prezentovať s čo najväčšou presnosťou a preto ich uvádzame v takejto podobe. Vidíme, že

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(1)}}, \boldsymbol{\theta})] = 3.8076 \nleq B(\pi^{(1)}) = 3.6906,$$

teda pravidlo (6) nie je splnené pre všetky  $\boldsymbol{\theta} \in \Theta$ . Ako sme vyššie spomenuli, body pridávané do návrhu vytvárali zhluky, ktoré môžeme vidieť na Obrázku 3. Kvôli lepšej vizualizácii sme váhy, s ktorými sme tieto body do návrhu pridávali, transformovali do logaritmickej škály.



**Obr. 3:** Pr. 2, H-algoritmus. Body pridávané do návrhu  $\xi^{\pi^{(1)}}$ 

Podľa Kroku 3 Podkapitoly 3.1 vygenerujeme 5 nových apriórnych rozdelení pomocou vyššie definovanej pomocnej siete. Najväčšia hodnota AVE kritéria patrí návrhu, ktorý je skonštruovaný pri apriórnom rozdelení pre t=1, teda  $\pi^{(2)} = \pi_1^{(2)}$ .

$$\pi^{(2)} = \begin{cases} 0.2, & \text{pre } (\theta_1, \theta_2) = (0.5, 4), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (0.5, 5), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (1.5, 4), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (1.5, 5), \\ 0.2, & \text{pre } (\theta_1, \theta_2) = (1, 5). \end{cases}$$

Približne AVE optimálny návrh, ktorý sme skonštruovali pri apriórnom rozdelení  $\pi^{(2)}$  spolu s ostatnými hodnotami prezentujeme v Tabuľke 9. Môžeme vidieť, že naše výsledky sú opäť podobné výsledkom autora Nyquista. Je však zrejmé, že v tejto tabuľke

#### 4.2 Príklad 2

(narozdiel od Tabuľky 8) sú naše výsledky viac vzdialené od Nyquistových. Pri porovnaní hodnoty AVE kritéria vidíme, že návrh, ktorý skonštruoval Nyquist, má túto hodnotu nižšiu, teda je lepší. Pokračujeme však v algoritme ďalej.

	naše výsledky	výsledky autora Nyquista		
¢π <sup>(2)</sup>	$\int 0.297  0.847  1.153  1.703 $	$\int 0.31  0.90  1.10  1.69$		
\$	0.252  0.266  0.227  0.255	$0.272  0.228  0.228  0.272 \int$		
$\epsilon^{\pi^{(2)}}$	0.000001	0.0017		
$\Delta_{min}^{\pi^{(2)}}$	0.2			
$B(\pi^{(2)})$	3.6966	3.6948		
$\boxed{\max_{\boldsymbol{\theta} \in \Theta} \Phi[\boldsymbol{M}(\xi^{\pi^{(2)}}, \boldsymbol{\theta})]}$	3.8164	3.7900		
$oldsymbol{ heta}^{(2)}=( heta_1, heta_2)$	(0.5, 5), (1.5, 5)	(0.5, 5), (1.5, 5)		
$pi^{\pi^{(2)}}$	1596			
$ au^{\pi^{(2)}}/ ext{sek.}$	232			

**Tabul'ka 9:** Pr. 2, H-algoritmus. Porovnanie výsledkov pre rozdelenie  $\pi^{(2)}$ 

Na Obrázku 4 vidíme graf bodov, ktoré sme pri konštrukcii návrhu  $\xi^{\pi^{(2)}}$ do návrhu postupne pridávali.



**Obr. 4:** Pr. 2, H-algoritmus. Body pridávané do návrhu $\xi^{\pi^{(2)}}$ 

Podmienka v Kroku 4, teda

$$B(\pi^{(2)}) = 3.6966 \ge B(\pi^{(1)}) = 3.6906,$$

je splnená a tak overíme nasledovné

$$\max_{\theta \in \Theta} \Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(2)}}, \boldsymbol{\theta})] = 3.8164 \nleq B(\pi^{(1)}) = 3.6966.$$

Pravidlo zastavenia (6) nie je splnená pre všetky  $\boldsymbol{\theta} \in \Theta$  a preto generujeme 5 nových apriórnych rozdelení tak, ako sme to popísali pri prezentovaní Nyquistových výsledkov. Hodnota AVE kritéria je najväčšia pre apriórne rozdelenie pre t=5, teda  $\pi^{(3)} = \pi_5^{(3)}$ , a má tvar

$$\pi^{(3)} = \begin{cases} 0.5, & \text{pre } (\theta_1, \theta_2) = (0.5, 5), \\ 0.5, & \text{pre } (\theta_1, \theta_2) = (1.5, 5). \end{cases}$$

V nasledujúcej tabuľke porovnáme výsledky, ku ktorým sme sa pre apri<br/>órne rozdelenie  $\pi^{(3)}$  dopracovali my a ku ktorým sa dopracoval autor Ny<br/>quist.

	naše výsledky	výsledky autora Nyquista
¢ $\pi^{(3)}$	$\int 0.287  1.000  1.713$	$\int 0.287  1.000  1.713$
5	$0.2708  0.4651  0.2641 \int$	$0.2675  0.4650  0.2675 \int$
$\epsilon^{\pi^{(3)}}$	0.0000005	0.0005
$\Delta_{min}^{\pi^{(3)}}$	0.4	
$B(\pi^{(3)})$	3.7729	3.7728
$\max_{\boldsymbol{\theta}\in\Theta}\Phi[\boldsymbol{M}(\boldsymbol{\xi}^{\pi^{(3)}},\boldsymbol{\theta})]$	3.7845	3.7728
$oldsymbol{ heta}^{(3)}=( heta_1, heta_2)$	(1.5, 5)	(1.5, 5)
$pi^{\pi^{(3)}}$	250	
$ au^{\pi^{(3)}}/ ext{sek.}$	14	

Tabuľka 10: Pr. 2, H-algoritmus. Porovnanie výsledkov pre rozdelenie  $\pi^{(3)}$ 

Na Obrázku 5 vidíme body, ktoré sme pri konštrukcii návrhu  $\xi^{\pi^{(3)}}$ do návrhu postupne pridávali.



**Obr. 5:** Pr. 2, H-algoritmus. Body pridávané do návrhu  $\xi^{\pi^{(3)}}$ 

Podmienka v Kroku 4, teda

$$B(\pi^{(3)}) = 3.7729 \ge B(\pi^{(2)}) = 3.6966$$

je splnená. Body v návrhu pre apriórne rozdelenie  $\pi^{(3)}$ , ktorý sme skonštruovali, sú totožné s Nyquistovými. Nami skonštruovaný približne AVE optimálny návrh sa od autorovho líši len vo váhach jednotlivých bodoch, aj to minimálne. Skonštruovali sme návrh, ktorého hodnota kritéria je v porovnaní s Nyquistovým návrhom väčšia len o 0.0001. Vidíme, že rozdiel je minimálny, no aj napriek tomu má veľký dopad. Zatiaľčo Nyquistov približne AVE optimálny návrh spĺňa pravidlo zastavenia (6), náš návrh toto pravidlo nespĺňa pre všetky  $\boldsymbol{\theta} \in \Theta$ , pretože vidíme, že

$$\max_{\theta \in \Theta} \Phi[\mathbf{M}(\xi^{\pi^{(3)}}, \theta)] = 3.7845 \nleq B(\pi^{(3)}) = 3.7729.$$

Keďže tento návrh nespĺňa pravidlo zastavenia, tak sme v algoritme pokračovali ďalej, no k približne minimaxnému optimálnemu návrhu sme sa nedopracovali. Je potrebné nájsť dôvod, prečo sa nám nepodarilo nájsť taký návrh, k akému sa dopracoval autor Nyquist, resp. návrh s takou hodnotou kritéria. Odpoveď na túto otázku sme načrtli už v predchádzajúcom texte. Problémom je spôsob, akým hľadáme približne AVE optimálny návrh. Keďže autor v článku [6] neuvádza spôsob konštruovania približne AVE optimálnych návrhov, pokúšali sme sa tieto návrhy konštruovať rôznymi spôsobmi. Pri hľadaní správneho postupu sme skúšali rôzne reštrikcie a vylepšenia. Hoci sme dospeli k veľmi podobným výsledkom ako sám autor Nyquist, nepodarilo sa nám dopracovať k rovnakým záverom ako autor, ktoré, ako sme ukázali na začiatku Podkapitoly 4.2.1, sú správne. Celkový čas výpočtu, ktorým sme sa dospracovali k návrhu  $\xi^{\pi^{(3)}}$ , je 27 minút a 29 sekúnd.

#### 4.2.2 Riešenie pomocou entropijnej regularizácie

Príklad 2 sme riešili aj pomocou entropijnej reguralizácie. V tejto časti analyzujeme výsledky dosiahnuté pri hodnotách parametra<sup>2</sup>  $\lambda = 0.1, 1, 10$  a 100. Pri konštruovaní približne entropijného optimálneho návrhu sme postupovali rovnako ako pri konštrukcii jednotlivých približne AVE optimálnych návrhov, teda body do návrhu sme pridávali symetricky tak, ako to opisujeme v Podkapitole 4.2.1. Tieto body sme do návrhu pridávali s krokom, ktorý minimalizuje hodnotu kritéria, teda s "optimálnym" krokom  $\alpha_n$  definovaným nasledovne

$$\alpha_n = \arg\min_{\alpha \in (0,1)} \Phi^e(\xi_{n+1}) = \arg\min_{\alpha \in (0,1)} \frac{1}{\lambda} ln \left[ \int_{\Theta} e^{\lambda \Phi[\boldsymbol{M}(\xi_{n+1}, \boldsymbol{\theta})]} d\boldsymbol{\theta} \right],$$

respektíve

$$\alpha_n = \arg\min_{\alpha \in (0,1)} \frac{1}{\lambda} ln \left[ \int_{\Theta} e^{\lambda \Phi [\boldsymbol{M}((1-\alpha)\xi_n + \frac{\alpha_n}{2}\delta_{x_{min1}}(x) + \frac{\alpha_n}{2}\delta_{x_{min2}}(x), \boldsymbol{\theta})]} d\boldsymbol{\theta} \right].$$

Pri konštrukcii približne entropijného optimálneho návrhu sme rovnako ako v Príklade 1 používali pravidlo zastavenia (14), ktorého detailné odvodenie uvádzame v Podkapitole 3.2.

<sup>&</sup>lt;sup>2</sup>Pre hodnoty parametra  $\lambda$  vyššieho rádu sa pri výpočtoch vyskytuje číslo väčšie ako číslo realmax=1.797693134862316 \* 10<sup>308</sup>, ktoré program Matlab vníma ako nekonečno.

V nasledujúcich tabuľkách prezentujeme výsledky, ku ktorým sme sa dopracovali pre jednotlivé hodnoty parametra  $\lambda$ . V tabuľkách uvádzame

- $\xi^e$  približne entropijný optimálny návrh,
- $\epsilon^e$  "vzdialenosť" návrhu $\xi^e$ od optimálneho návrhu,
- $\Delta^e_{min}$ minimálna vzájomná vzdialenosť 2 bodov (popisané v Kapitole 2),
- $\Phi^e(\xi^e)$  hodnota entropijného kritéria návrhu  $\xi^e$ ,
- $\Phi^m(\xi^e)$  hodnota minimaxného kritéria návrhu  $\xi^e$ ,
- $\boldsymbol{\theta}^e$  bod, v ktorom sa nadobúda  $\Phi^m(\xi^e)$ ,
- $pi^e$  počet iterácii pri konštrukcii návrhu  $\xi^e$ ,
- $\tau^e$  čas výpočtu pri konštrukcii návrhu  $\xi^e$ .

Tabuľka 11 obsahuje výsledky pre hodnoty parametra  $\lambda$  rovné 0.1 a 1, v Tabuľke 12 uvádzame hodnoty výsledkov pre parameter  $\lambda$  rovný 10 a 100.

	parameter $\lambda = 0.1$	parameter $\lambda = 1$
¢e	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	$\int 0.457  0.997  1.543$
ς	$0.3456 \ 0.3088 \ 0.3456 $	$0.3325 \ 0.3350 \ 0.3325 $
$\epsilon^e$	0.0005	0.0367
$\Delta^e_{min}$	0.3	0.3
$\Phi^e(\xi^e)$	3.4166	3.4363
$\Phi^m(\xi^e)$	4.0335	3.9722
$oldsymbol{ heta}^e=( heta_1, heta_2)$	(1.5, 5)	(1.5, 5)
$pi^e$	78	76
$ au^e$	3h 36m 24s	3h 37m 12s

**Tabuľka 11:** Pr. 2, entropijná regularizácia. Porovnanie výsledkov pre hodnotu parametra  $\lambda = 0.1, 1$ 

Z uvedených výsledkov vidíme, podobne ako v Príklade 1, že so zvyšujúcou sa hodnotou parametra  $\lambda$  hodnota entropijného kritéria približne optimálneho návrhu stúpa

	parameter $\lambda = 10$	parameter $\lambda = 100$
çe	$\int 0.352  1.007  1.648$	$\int 0.254  0.873  1.270  1.746$
ς	$0.2903  0.4154  0.2943 \int$	$0.2563  0.3149  0.1720  0.2568 \int$
$\epsilon^e$	0.1003	0.7868
$\Delta^e_{min}$	0.3	0.3
$\Phi^e(\xi^e)$	3.5340	3.7462
$\Phi^m(\xi^e)$	3.8215	3.8149
$oldsymbol{ heta}^e = ( heta_1,  heta_2)$	(0.5, 5)	(0.5, 5)
$pi^e$	24	66
$ au^e/{ m min.}$	1h 33m 28s	6h 17m 31s

**Tabuľka 12:** Pr. 2, entropijná regularizácia. Porovnanie výsledkov pre hodnotu parametra  $\lambda = 10,100$ 

a hodnota minimaxného kritéria klesá. Obrázok 6 graficky znázorňuje tento postreh. Najnižšiu hodnotu minimaxného kritéria má návrh, ktorý sme skonštru<br/>ovali pri parametri  $\lambda = 100$ a tou hodnotou je číslo 3.8149. Autor článku [6] skonštru<br/>oval pomocou H-algoritmu návrh s hodnotou minimaxného kritéria 3.7728, teda "optimálnejší" návrh.



**Obr. 6:** Pr. 2, entropijná regularizácia. Graf hodnôt kritérií v závislosti od parametra  $\lambda$ 

Na záver prikladáme grafické znázornenie bodov pridávaných do návrhu pre jednotlivé hodnoty parametra  $\lambda$ . Kvôli lepšej vizualizácii sme váhy, s ktorými sme tieto body do návrhu pridávali, transformovali do logaritmickej škály. Podobne ako pri riešení príkladu pomocou H-algoritmom sme v grafe zaznačili zhluky, ktoré tieto body vytvárajú.



**Obr. 7:** Pr. 2, entropijná regularizácia. Body pridávané do návrhu pre jednotlivé hodnoty parametra  $\lambda$ 

# 5 Diskusia k riešeným príkladom

Pri riešení príkladov sme dospeli k výsledkom, ktoré chceme vyhodnotiť a porovnať práve v tejto kapitole s názvom Diskusia k riešeným príkladom. V prvom rade sa budeme venovať výsledkom a záverom Príkladu 1. Tak ako uvádzame v Podkapitole 4.1, tento príklad je jednoparametrický a pri riešení pomocou oboch prezentovaných prístupov sme dospeli k jednobodovému optimálnemu návrhu.

V Podkapitole 4.1.1 popisujeme riešenie dosiahnuté H-algoritmom, kde porovnávame naše výsledky spolu s výsledkami autora Nyquista. Ako môžeme vidieť, nami zostrojený minimaxný optimálny návrh má tvar

$$\xi^m = \begin{cases} 1.9999\\ 1 \end{cases}$$

a hodnota minimaxného kritéria optimálneho návrhu je  $\Phi^m(\xi^m) = 63.995769224423420$ . Nyquistov minimaxný optimálny návrh vyzerá nasledovne

$$\xi^m = \begin{cases} 2\\ 1 \end{cases}$$

a hodnota minimaxného kritéria, ktoré uvádza autor je  $\Phi^m(\xi^m) = 64.0000$ . Je jasné, že naše výsledky by boli po zaokrúhlení totožné s Nyquistovými, teda môžeme konštatovať, že sme spolu s autorom Nyquistom dospeli k rovnakým záverom. Chceme poukázať na malú chybu, na ktorú sme pri kontrole Nyquistových výsledkoch narazili. Medzivýsledky, ktoré autor Nyquist uvádza v článku [6] v Table 1 *(Results from first iteration in Example 1)* sú pre t=1, 2 a 3 nesprávne, pretože autor skonštruoval návrhy, ktoré nie sú AVE optimálne. Hodnoty AVE kritéria týchto návrhov sú väčšie ako uvádza autor. Podrobnejšiu analýzu uvádzame v Podkapitole 4.1.1.

Pri konštrukcii minimaxného optimálneho návrhu pomocou entropijnej regularizácie sme analyzovali výsledky dosiahnuté pre hodnotu parametra  $\lambda$ =0.1, 1, 2, 5 a 8. Pri zvyšovaní hodnoty parametra  $\lambda$  hodnota entropijného kritéria nami skonštruovaného optimálneho návrhu stúpa a hodnota minimaxného kritéria klesá, teda hodnoty oboch kritérií sa k sebe približujú. Tento postreh je graficky znázornený na Obrázku 2. So stúpajúcou hodnotou parametra  $\lambda$  je skonštruovaný entropijný optimálny návrh bližšie k minimaxnému optimálnemu návrhu, ku ktorému sme dospeli pomocou H-algoritmu, čo môžeme graficky vidieť na Obrázku 1. Entropijný optimálny návrh s najnižšou hodnotou minimaxného kritéria je návrh skonštruovaný pri hodnote parametra  $\lambda=8$  a má tvar

$$\xi^e = \left\{ \begin{array}{c} 1.9980431594\\ 1 \end{array} \right\}$$

Hodnota minimaxného kritéria tohto návrhu je  $\Phi^m(\xi^e) = 63.995797850449875.$ 

Ako vidíme, rozdiely dosiahnutých výsledkov pomocou oboch prístupov sú minimálne, no nižšia hodnota minimaxného kritéria patrí návrhu skonštruovanému pomocou H-algoritmu. Preto musíme konštatovať, že H-algoritmom sme dosiahli efektívnejšie výsledky. Pri entropijnej regularizácii nás obmedzovala voľba hodnoty parametra  $\lambda$ . Pri výpočtoch sa v tomto príklade pre hodnotu  $\lambda > 8$  vyskytuje číslo väčšie ako povolené maximálne reálne číslo v programe Matlab. Časová náročnosť výpočtu sa pomocou entropijnej regularizácie pohybuje v desatinách sekundy, zatiaľčo H-algoritmom sme minimaxný optimálny návrh skonštruovali za necelé dve stotiny sekundy.

Príklad 2 podrobne rozoberáme v Kapitole 4.2. Keďže to je príklad logistickej regresie, tak sme v úvode tejto podkapitoly predostreli krátky úvodný komentár, v ktorom popisujeme tvar informačnej matice.

V prvej časti prezentujeme výsledky, ku ktorým sa pomocou H-algoritmu dopracoval autor Nyquist. Tieto výsledky sme podrobne skontrolovali a konštatujeme, že sú správne. V druhej časti tejto podkapitoly uvádzame naše výsledky dosiahnuté pomocou H-algoritmu. Nepodarilo sa nám skonštruovať približne minimaxný optimálny návrh. Výsledky, ktoré sme v medzivýpočtoch dosiahli, boli veľmi blízke výsledkom autora Nyquista. Približne minimaxný optimálny návrh [6] skonštruovaný pomocou H-algoritmu je

$$\xi^m = \begin{cases} 0.287 & 1.00 & 1.713\\ 0.2675 & 0.4650 & 0.2675 \end{cases}$$

Hodnota minimaxného kritéria je  $\Phi^m(\xi^m) = 3.7728$  a "najnepriaznivejšie rozdelenie" má tvar

$$\pi^{(3)} = \begin{cases} 0.5, & \text{pre } (\theta_1, \theta_2) = (0.5, 5), \\ 0.5, & \text{pre } (\theta_1, \theta_2) = (1.5, 5). \end{cases}$$

Nami skonštruovaný návrh pri tomto rozdelení má hodnotu minimaxného kritéria

 $\Phi^m(\xi^{\pi^{(3)}})=3.7845$ a jeho tvar je nasledovný

$$\xi^{\pi^{(3)}} = \begin{cases} 0.287 & 1.000 & 1.713 \\ 0.2708 & 0.4651 & 0.2641 \end{cases}.$$

Tento návrh nesplňa pravidlo zastavenia. Problém je v konštrukcii jednotlivých približne AVE optimálnych návrhov. Autor Nyquist sa v článku [6], v ktorom prezentuje H-algoritmus, vôbec nezmieňuje a ani len nenačrtne, akým spôsobom konštruuje približne AVE optimálne návrhy v jednotlivých iteráciách. Keďže konštrukcia približne AVE optimálneho návrhu je v tomto algoritme veľmi dôležitá, tak si myslíme, že sa autor článku mal vyjadriť aj k tejto časti. Takisto by sme sa chceli kriticky vyjadriť ku skutočnosti, ako sa v teoretickej časti článku píše, že jedným z predpokladov je kompaktnosť množiny  $\mathscr{X}$ , no v samotných príkladoch autor túto množinu neuvádza.

V tretej časti Podkapitoly 4.2 riešime a analyzujeme výsledky dosiahnuté entropijnou regularizáciou pre hodnoty parametra  $\lambda$ =0.1, 1, 10 a 100. Pre hodnoty vyššieho rádu sa opäť vyskytuje číslo väčšie ako je v programe Matlab povolené. Dosiahnuté výsledky mali rovnaký charakter ako v Príklade 1, teda so zvyšujúcou sa hodnotou parametra  $\lambda$  hodnota entropijného kritéria približne optimálneho návrhu stúpa a hodnota minimaxného kritéria klesá, teda opäť sa tieto hodnoty k sebe približujú. Tento fakt môžeme vidieť v grafickej podobne na Obrázku 6. Minimálna hodnota minimaxného kritéria  $\Phi^m(\xi^e) = 3.8149$  patrí návrhu skonštruovanému pri hodnote parametra  $\lambda = 100$  a má tvar

$$\xi^{e} = \left\{ \begin{array}{rrrr} 0.254 & 0.873 & 1.270 & 1.746 \\ 0.2563 & 0.3149 & 0.1720 & 0.2568 \end{array} \right\}$$

Môžeme vidieť, že hodnota minimaxného kritéria približne optimálneho návrhu zostrojeného H-algoritmom je nižšia ako hodnota minimaxného kritéria nájdeného návrhu, ktorý sme skonštruovali pomocou entropijnej regularizácie. H-algoritmus sa v tomto prípade javí ako efektívnejší prístup konštruovania približne minimaxného optimálneho návrhu. Porovnávať časovú náročnosť výpočtov je v tomto prípade ťažké, pretože H-algoritmom sa nám nepodarilo nájsť približne minimaxný optimálny návrh. Ako sme spomenuli, naše výsledky boli veľmi blízko správnym výsledkom, tak môžeme porovnávať čas, ktorý bol potrebný na skonštruovanie návrhu  $\xi^{\pi^{(3)}}$ . V tomto prípade sa ako efektívnejší prístup javí opäť H-algoritmus, keďže výpočty trvajú rádovo v minutách, zatiaľčo výpočty entropijnou regularizáciou v hodinách. Pri entropijnej regularizácii je časovo náročný výpočet dvojrozmerného integrálu. V programe Matlab je integrovaná funkcia s názvom *integral2*, ktorá rieši dvojrozmerný integrál numerickými metódami. V našom prípade nebolo možné použiť túto funkciu, pretože v medzivýpočtoch pracujeme s maticou a táto funkcia si s touto skutočnosťou poradiť nevie. Vnoreným prístupom sme použili funkciu *integral*, ktorá slúži na výpočet jednorozmerného integrálu. Dôvod, prečo sme použili túto funkciu je ten, že obsahom funkcie *integral* je možnosť '*ArrayValued*', pomocou ktorej si Matlab vie poradiť s tým, že v medzivýpočtoch pracujeme s maticou. Argument '*ArrayValued*'=true vo funkcii zabezpečí to, že numerický výpočet integrálu nie je realizovaný pomocou "skalárneho súčinu vektorov" (nemožné kvôli výskytu matice), ale "for cyklom". Práve táto skutočnosť je dôvod, prečo je výpočet dvojného integrálu, teda aj samotného algoritmu, časovo tak náročný.

## Záver

V diplomovej práci sme sa zaoberali minimaxnými optimálnymi návrhmi regresných experimentov. Cieľom práce bolo, v nadväznosti na existujúce prednášky [8], oboznámiť sa v literatúre s existujúcimi prístupmi a na počítači preveriť nedávno publikovanú metódu Nyquista na konkrétnych úlohách.

Prvá kapitola obsahuje lokálne kritéria optimality, kritéria optimality v nelineárnom modeli ako aj nutné a postačujúce podmienky optimality. V Kapitole 2 je rozpracovaná jedna numerická metóda hľadania AVE optimálneho návrhu. V poslednej teoretickej kapitole s názvom Opis numerických metód hľadania minimaxného optimálneho návrhu sme podrobne rozpracovali H-algoritmus [6] a entropijnú regularizáciu minimaxného kritéria. Pri samotnom H-algoritme sme uviedli podrobnejší dôkaz Vety 3.1 ako uvádza sám autor algoritmu. Ukázali sme, že v prípade konečnosti množín  $\mathscr{X}$  a  $\Theta$  je Veta 3.1, ako vyplýva z Vety o minmaxe [3], nutnou i postačujúcou podmienkou optimality. Detailnejšie sme odvodili nutné a postačujúce podmienky optimality a pravidlo zastavenia v prípade entropijnej regularizácie.

V Kapitole 4 sme prezentovali výsledky príkladov dosiahnuté pomocou H-algoritmu a entropijnou regularizáciou. Uviedli sme podrobné analýzy, tabuľky a grafy. V poslednej kapitole sme tieto výsledky navzájom porovnali a analyzovali. V Príklade 1 boli výsledky dosiahnuté oboma prístupmi po zaokrúhlení totožné. Výsledky dosiahnuté H-algoritmom boli v Príklade 2 z hľadiska optimality efektívnejšie ako výsledky dosiahnuté entropijnou regularizáciou, i keď rozdiely boli minimálne. Približne minimaxný optimálny návrh [6] skonštruovaný H-algoritmom má hodnotu minimaxného kritéria  $\Phi^m(\xi^m) = 3.7728$  a návrh skonštruovaný entropijnou regularizáciou pri hodnote parametra  $\lambda = 100 \ \Phi^m(\xi^e) = 3.8149$ . Časová náročnosť výpočtov pomocou H-algoritmu bola rádovo v minutách, zatiaľčo výpočty entropijnou regularizáciou v hodinách a tak konštatujeme, že H-algoritmus je aj z hľadiska časovej náročnosti efektívnejší.

## Zoznam použitej literatúry

- Atkinson, A. C., Donev, A. N., Tobias, R. D.: Optimum Experimental Designs, with SAS, Oxford University Press, Oxford, 2007
- [2] Fedorov, V. V.: Theory of Optimal Experiments, Academic Press, New York, 1972
- [3] Hamala, M., Trnovská, M.: Nelineárne programovanie, Epos, Bratislava, 2013
- [4] Müller, CH. H., Pázman, A.: Application of necessary and sufficient conditions for maximin efficient design, Metrika 48, 1-19, Springer-Verlag, 1998
- [5] Nyquist, H.: Conditions for minimax designs, Working Paper, Stockholm University, Stockholm, 2012
- [6] Nyquist, H.: Convergence of an Algorithm for Constructing Minimax Design. In Ucinsky, D., Atkinson, A. C., Patan, M. (eds.), mODa10-Advances on Model-Oriented Design and Analysis, 169-176, Physica-Verlag, Heidelberg, 2013
- [7] Pázman, A.: Fundations of Optimum Experimental Design, Reidel (Kluwer Group), Dordrecht, 1986
- [8] Pázman, A., Lacko, V.: Prednášky z regresných modelov, Univerzita Komenského v Bratislave, Bratislava, 2012
- [9] Pronzato, L., Pázman, A.: Design of Experiments in Nonlinear Models, Springer, New York, 2013
- [10] Pukelsheim, F.: Optimal Design of Experiments, John Wiley & Sons, Inc., New York, 1993
- [11] Wong, W. K., Berger, M. P. F.: An Introduction to Optimal Designs for Social and Biomedical Research, John Wiley & Sons, Inc., Chichester, 2009

# Príloha A

Príklad 1, H-algoritmus\ Matlab R2013a

```
function [efko]=f(th,xi,i,j)
  efko=[-xi(1,i)/(th(j,1)+xi(1,i))^2]';
end
function [efko_min]=f_min(th,x,j)
  efko_min=[-x/(th(j,1)+x)^2]';
end
function [efko]=f_Theta(t,xi_AVE,i)
  efko=[-xi_AVE(1,i)/(t+xi_AVE(1,i))^2]';
end
function [G]=gradient(M)
 G = -M^{(-2)};
end
function [K]=kriterium(M)
 K=M^{(-1)};
end
function [Kriterium] = Max_Th(xi_AVE,t)
 size_xi_AVE=size(xi_AVE);
 M_Theta=0;
  for k=1:size_xi_AVE(2)
      M_Theta=M_Theta+xi_AVE(2,k)*f_Theta(t,xi_AVE,k)*f_Theta(t,xi_AVE,k)';
  end
  Kriterium=kriterium(M_Theta);
end
function [E_pi_c,Krit]=Epic(xi,th)
%funkcia na vypocet E_pi_c
 size_xi=size(xi);
 size_th=size(th);
 E_pi_c=0;
  for i=1:size xi(2)
      I=0;
      Krit=0;
      for j=1:size_th(1)
```

```
M=0;
          for k=1:size_xi(2)
              M=M+xi(2,k)*f(th,xi,k,j)*f(th,xi,k,j)';
          end
          G=gradient(M);
          I=I+th(j,size_th(2))*f(th,xi,i,j)'*G*f(th,xi,i,j);
          Krit=Krit+th(j,size_th(2))*kriterium(M);
      end
      E_pi_c=E_pi_c+xi(2,i)*I;
  end
end
function [E_pi]=Epi(x,th,xi)
%funkcia na vypocet E_pi
 size_th=size(th);
 size_xi=size(xi);
 E_pi=0;
  for j=1:size_th(1)
     M=0;
      for k=1:size_xi(2)
          M=M+xi(2,k)*f(th,xi,k,j)*f(th,xi,k,j)';
      end
      Grad=gradient(M);
      E_pi=E_pi+th(j,size_th(2))*f_min(th,x,j)'*Grad*f_min(th,x,j);
 end
end
function [Krit,epsilon]=ave_optimalita(xi,th,X)
%funkcia na overenie AVE optimality
  [E_pi_c,Krit]=Epic(xi,th);
  [x_min, minimum] = fminbnd(@(x)Epi(x,th,xi),X(1),X(2));
 epsilon=E_pi_c-minimum;
end
function [hodnota,theta]=minimax_kriterium(xi,Th)
%funkcia na vypocet minimax kriteria
  [theta value]=fminbnd(@(t1)-minimax(xi,t1),Th(1),Th(2));
 hodnota=-value;
end
function [xi,Krit,rozdiel]=ave_algoritmus(X,th,eps)
%funkcia AVE algoritmu
```

```
tic
 xi=[X(1);1];
 E_pi_c=0;
 E_pi=0;
 minimum=Inf;
 rozdiel=[];
 x_min=[];
 poc=0;
  [E_pi_c,Krit]=Epic(xi,th);
  [x_min, minimum] = fminbnd(@(x)Epi(x,th,xi),X(1),X(2));
 while (abs(minimum-E_pi_c)>eps)
      poc=poc+1;
      size_xi=size(xi);
      size_th=size(th);
      E_pi_c=0;
      [E_pi_c,Krit]=Epic(xi,th);
      [x_min, minimum] = fminbnd(@(x)Epi(x,th,xi),X(1),X(2));
      xi=[x_min;1];
      rozdiel=[rozdiel;abs(minimum-E_pi_c)];
 end
 toc;
end
function ...
   [xi_MinMax,pi_MinMax,Maximum,B,cas]=H_algoritmus(X,Theta,h_t,eps)
%funkcia H-algoritmu
 tic
 B=-Inf;
 size_Theta=size(Theta);
 number_parameters=size_Theta(1,1);
 pi=[Theta(1,1),1];
 Maximum=Inf;
 B=0;
 while (Maximum>B)
      [xi_AVE,Krit,rozdiel]=ave_algoritmus(X,pi,eps);
      B=Krit;
      xi_MinMax=xi_AVE;
      pi_MinMax=pi;
      size_xi_AVE=size(xi_AVE);
```

```
Max_Theta=[];
      [Max_Theta,value]=fminbnd(@(t)-Max_Th(xi_AVE,t),Theta(1),Theta(2));
      Maximum=-value;
      if B==Maximum
          break
      end
      size_Max_Theta=size(Max_Theta);
      size_pi=size(pi);
      max_krit=-Inf;
      for h=h_t:h_t:1
          Pi=pi;
          Pi(:,size_pi(1,2))=(1-h)*Pi(:,size_pi(1,2));
          Pi=[Pi;Max_Theta, (h/size_Max_Theta(1,1)) * ones (size_Max_Theta(1,1),
          1)];
          [xi_AVE,Krit,rozdiel]=ave_algoritmus(X,Pi,eps);
          if (Krit>max_krit)
              max_krit=Krit;
              rozdelenie=Pi;
              krok=h;
              xi_H=xi_AVE;
          end
      end
      pi=rozdelenie;
 end
 cas=toc;
end
```

#### Príklad 1, entropijná regularizácia\ Matlab R2013a

```
function [efko]=f_Theta(t,xi,i)
    efko=[-xi(1,i)/(t+xi(1,i))^2]';
end
function [efko]=f_Theta_min(t,x)
    efko=[-x/(t+x)^2]';
end
function [G]=gradient(M)
```

```
G = -M^{(-2)};
end
function [K]=kriterium(M)
 K=M^{(-1)};
end
function [I]=Int_c(xi,t,lambda)
% funkcia na rátanie Integrálu INT(e*c)
  size_xi=size(xi);
 C=0;
 for i=1:size_xi(2)
      M=0;
      for k=1:size_xi(2)
          M=M+xi(2,k)*f_Theta(t,xi,k)*f_Theta(t,xi,k)';
      end
      G=gradient(M);
      Krit=kriterium(M);
      C=C+xi(2,i)*f_Theta(t,xi,i)'*G*f_Theta(t,xi,i);
  end
  I=exp(lambda*Krit)*C;
end
function [I]=Int_fgf(xi,x,t,lambda)
% funkcia na rátanie Integrálu INT(e*fgf)
 size_xi=size(xi);
 C=0;
 M=0;
      for k=1:size_xi(2)
          M=M+xi(2,k)*f_Theta(t,xi,k)*f_Theta(t,xi,k)';
      end
 G=gradient(M);
 Krit=kriterium(M);
 I=exp(lambda*Krit)*f_Theta_min(t,x)'*G*f_Theta_min(t,x);
end
function [En_krit,epsilon]=Entropia_optimalita(X,xi,th,lambda)
%funkcia na vypocet Entr. kriteria a overenie optimality
  I_c=integral(@(t)Int_c(xi,t,lambda),th(1),th(2),'ArrayValued',true);
  [x_min,I_fgf]=fminbnd(@(x)integral(@(t)Int_fgf(xi,x,t,lambda),th(1),th(2),
  'ArrayValued',true),X(1),X(2));
  epsilon=abs(I_c-I_fgf);
```

```
En_krit_pom=integral(@(t)Entr_krit(xi,t,lambda),th(1),th(2),
  'ArrayValued',true);
 En krit=(1/lambda) *log(En krit pom);
end
function [I]=Entr_krit(xi,t,lambda)
%pomocna funkcia na vypocet Entr. kriteria
  size_xi=size(xi);
 M=0;
      for k=1:size_xi(2)
          M=M+xi(2,k)*f_Theta(t,xi,k)*f_Theta(t,xi,k)';
      end
 Krit=kriterium(M);
  I=exp(lambda*Krit);
end
function ...
   [xi,XMIN,EN_KRIT,epsilon,m,cas]=Entropia_algoritmus(X,th,lambda,eps)
%funkcia Entropijneho algoritmu
 tic
 xi=[X(1);1];
 En_krit_pom=integral(@(t)Entr_krit(xi,t,lambda),th(1),th(2),'ArrayValued',
 true);
 En_krit=(1/lambda) *log(En_krit_pom);
 EN_KRIT=[];
 EN_KRIT=[EN_KRIT;En_krit];
  I_c=integral(@(t)Int_c(xi,t,lambda),th(1),th(2),'ArrayValued',true);
  [x_min,I_fqf]=fminbnd(@(x)integral(@(t)Int_fqf(xi,x,t,lambda),th(1),th(2),
  'ArrayValued',true),X(1),X(2));
 XMIN=[];
 XMIN=[XMIN; x_min]
 m=1;
 epsilon=[];
  epsilon=[epsilon,abs(I_c-I_fgf)/En_krit_pom];
  while (abs(I_c-I_fgf)/En_krit_pom>eps)
     m = m + 1;
      xi=[x_min;1];
      En_krit_pom=integral(@(t)Entr_krit(xi,t,lambda),th(1),th(2),
      'ArrayValued',true);
      En_krit=(1/lambda) *log(En_krit_pom);
```

```
EN_KRIT=[EN_KRIT;En_krit];
I_c=integral(@(t)Int_c(xi,t,lambda),th(1),th(2),'ArrayValued',true);
[x_min,I_fgf]=fminbnd(@(x)integral(@(t)Int_fgf(xi,x,t,lambda),th(1),
th(2),'ArrayValued',true),X(1),X(2));
XMIN=[XMIN;x_min];
epsilon=[epsilon,abs(I_c-I_fgf)/En_krit_pom];
end
cas=toc;
end
```

Príklad 2, H-algoritmus \ Matlab R2013a

```
function [efko]=f(th,xi,i,j)
  w = (\exp(th(j,2) * (xi(1,i)-th(j,1))) / (1+\exp(th(j,2) * (xi(1,i)-th(j,1))))^2);
  efko=sqrt(w)*[-th(j,2),xi(1,i)-th(j,1)]';
end
function [efko_min]=f_min(th,x,j)
  w = (\exp(th(j,2) * (x-th(j,1))) / (1+\exp(th(j,2) * (x-th(j,1))))^{2});
  efko_min=sqrt(w) * [-th(j,2), x-th(j,1)]';
end
function [efko]=f_Theta(t1,t2,xi_AVE,i)
  w=(exp(t2*(xi_AVE(1,i)-t1))/(1+exp(t2*(xi_AVE(1,i)-t1)))^2);
  efko=sqrt(w) * [-t2, xi_AVE(1, i)-t1]';
end
function [G]=gradient(M)
  G=-M^{(-1)};
end
function [K]=kriterium(M)
  K=-\log(\det(M));
end
function [Kriterium] = Max_Th(theta, xi_AVE)
  size_xi_AVE=size(xi_AVE);
  t1=theta(1);
  t2=theta(2);
  M_Theta=0;
  for k=1:size_xi_AVE(2)
```

```
M_Theta=M_Theta+xi_AVE(2,k)*f_Theta(t1,t2,xi_AVE,k)*f_Theta
      (t1,t2,xi_AVE,k)';
  end
 Kriterium=kriterium(M_Theta);
end
function [E_pi_c,Krit]=Epic(xi,th)
%funkcia na vypocet E_pi_c
  size_xi=size(xi);
 size_th=size(th);
 E_pi_c=0;
  for i=1:size_xi(2)
      I=0;
      Krit=0;
      for j=1:size_th(1)
          M=0;
          for k=1:size_xi(2)
            M=M+xi(2,k)*f(th,xi,k,j)*f(th,xi,k,j)';
          end
          G=gradient(M);
          I=I+th(j,size_th(2))*f(th,xi,i,j)'*G*f(th,xi,i,j);
          Krit=Krit+th(j,size_th(2))*kriterium(M);
      end
      E_pi_c=E_pi_c+xi(2,i)*I;
  end
end
function [E_pi]=Epi(x,th,xi)
%funkcia na vypocet E_pi
 size_th=size(th);
 size_xi=size(xi);
 E_pi=0;
  for j=1:size_th(1)
      M=0;
      for k=1:size_xi(2)
          M=M+xi(2,k)*f(th,xi,k,j)*f(th,xi,k,j)';
      end
      Grad=gradient(M);
      E_pi=E_pi+th(j,size_th(2))*f_min(th,x,j)'*Grad*f_min(th,x,j);
  end
```

```
end
function [Krit,epsilon]=ave_optimalita(xi,th,X)
%funkcia na overenie AVE optimality
  [E_pi_c,Krit]=Epic(xi,th);
  [x_min, minimum] = fminbnd(@(x)Epi(x,th,xi),X(1),X(2));
  epsilon=abs(E_pi_c-minimum);
end
function [Maximum, Max_Theta] = minimax_kriterium(Theta, xi_AVE)
%funkcia na vypocet minimax kriteria
  size_xi_AVE=size(xi_AVE);
 Maximum=-Inf;
 Max_Theta=[];
  for t1=Theta(1,1):0.1:Theta(1,2)
      for t2=Theta(2,1):0.1:Theta(2,2)
          M_Theta=0;
          for k=1:size_xi_AVE(2)
              M_Theta=M_Theta+xi_AVE(2,k)*f_Theta(t1,t2,xi_AVE,k)*f_Theta
               (t1,t2,xi_AVE,k)';
          end
          Kriterium=kriterium(M_Theta);
          if (Kriterium==Maximum)
              Maximum=Kriterium;
              Max_Theta=[Max_Theta;t1,t2];
          end
          if (Kriterium>Maximum)
              Maximum=Kriterium;
              Max_Theta=[t1,t2];
          end
      end
  end
end
function [ALPHA]=krok(xi,th,x_min_1,x_min_2,alpha)
%funkcia na vypocet optimalneho kroku
  size_th=size(th);
  size_xi=size(xi);
 ALPHA=0;
  for j=1:size_th(1)
      M=0;
```
```
for k=1:size_xi(2)
          M=M+xi(2,k)*f(th,xi,k,j)*f(th,xi,k,j)';
      end
      M_alpha=(1-alpha) *M+(alpha/2) *f_min(th,x_min_1,j) *f_min(th,x_min_1,j) '
      +(alpha/2)*f_min(th,x_min_2,j)*f_min(th,x_min_2,j)';
      ALPHA=ALPHA+th(j,size_th(2))*kriterium(M_alpha);
 end
end
function ...
   [B,XMIN,KROK,xi_AVE,Krit,epsilon,m,cas]=ave_algoritmus(X,th,eps,vaha)
%funkcia AVE algoritmu
 tic
 xi = [X(1), X(1)/2 + X(2)/2, X(2); 1/3, 1/3, 1/3];
  [x_min_1, minimum] = fminbnd(@(x)Epi(x,th,xi),X(1),X(2));
 distance=x_min_1-X(1);
 x_min_2=X(2)-distance;
 XMIN=[];
 XMIN=[XMIN, x_min_1, x_min_2];
  [alpha, value]=fminbnd(@(alpha)krok(xi,th,x_min_1,x_min_2,alpha),0,1);
 KROK=[];
 KROK=[KROK, alpha/2, alpha/2];
 xi=[[x_min_1;alpha/2], [X(1)/2+X(2)/2;(1-alpha)], [x_min_2;alpha/2]];
  [E_pi_c]=Epic(xi,th);
 epsilon=[minimum-E_pi_c];
 B=[];
 m=1;
 k=1;
  while (abs(minimum-E_pi_c)>eps)
      m = m + 1;
      size_xi=size(xi);
      n=size xi(2);
      [alpha, ...
         value]=fminbnd(@(alpha)krok(xi,th,x_min_1,x_min_2,alpha),0,1);
      distance=x_min_1-X(1);
      x_min_2=X(2)-distance;
      XMIN=[XMIN, x_min_1, x_min_2];
      KROK=[KROK, alpha/2, alpha/2];
      xi=[xi(1,:);(1-alpha) *xi(2,:)];
```

```
xi=[xi,[x_min_1;alpha/2],[x_min_2;alpha/2]];
      for i=1:n+1
          for j=1:n+1
              if(i≠j & abs(xi(1,i)-xi(1,j))≤vaha)
              xi(1,i) = (xi(1,i) + xi(1,j)) / (2);
              xi(2,i)=xi(2,i)+xi(2,j);
              xi(:,[j])=[];
              xi=[xi,[Inf;Inf]];
              end
          end
      end
      a=[];
      for i=1:n+1
          if (xi(1,i)≠Inf)
              a=[a,xi(:,i)];
          end
      end
      b=sum(a(2,:));
      a=[a(1,:);(1/b)*a(2,:)];
      xi=a;
      [x_min_1, minimum] = fminbnd(@(x) Epi(x, th, xi), X(1), X(2));
      [E_pi_c,Krit]=Epic(xi,th);
      epsilon=[epsilon;minimum-E_pi_c];
      xi_AVE=xi;
      B=[B;Krit];
  end
  size_XMIN=size(XMIN);
  XMIN=[XMIN(1:2), XMIN(5:(size_XMIN(1,2)))];
  size_KROK=size(KROK);
  KROK=[KROK(1:2),KROK(5:(size_KROK(1,2)))];
  cas=toc;
end
```

## Príklad 2, entropijná regularizácia\ Matlab R2013a

function [efko]=f\_Theta(t1,t2,xi,i)

```
w = (\exp(t2 * (xi(1,i)-t1)) / (1+\exp(t2 * (xi(1,i)-t1)))^2);
  efko=sqrt(w) * [-t2, (xi(1,i)-t1)]';
end
function [efko]=f_Theta_min(t1,t2,x)
  w = (exp(t2*(x-t1))/(1+exp(t2*(x-t1)))^2);
 efko=sqrt(w) * [-t2, x-t1]';
end
function [G]=gradient(M)
 G = -M^{(-1)};
end
function [K]=kriterium(M)
 K=-\log(\det(M));
end
function [I]=Int_c(xi,t1,t2,lambda)
% funkcia na rátanie Integrálu INT(e*c)
 size_xi=size(xi);
 C=0;
  for i=1:size_xi(2)
      M=0;
      for k=1:size_xi(2)
          M=M+xi(2,k)*f_Theta(t1,t2,xi,k)*f_Theta(t1,t2,xi,k)';
      end
      G=gradient(M);
      Krit=kriterium(M);
      C=C+xi(2,i)*f_Theta(t1,t2,xi,i)'*G*f_Theta(t1,t2,xi,i);
  end
  I=exp(lambda*Krit)*C;
end
function [I]=Int_fgf(xi,x,t1,t2,lambda)
% funkcia na rátanie Integrálu INT(e*fgf)
 size xi=size(xi);
 M=0;
  for k=1:size_xi(2)
      M=M+xi(2,k)*f_Theta(t1,t2,xi,k)*f_Theta(t1,t2,xi,k)';
  end
  G=gradient(M);
  Krit=kriterium(M);
  I=exp(lambda*Krit)*f_Theta_min(t1,t2,x)'*G*f_Theta_min(t1,t2,x);
```

```
end
function [En_krit,epsilon]=Entr_optimalita(xi,X,th,lambda)
%funkcia na vypocet Entr. kriteria a overenie optimality
  I_c=integral(@(t2)integral(@(t1)Int_c(xi,t1,t2,lambda),th(1,1),th(1,2),
  'ArrayValued', true), th(2,1), th(2,2), 'ArrayValued', true);
  [x_min,I_fgf]=fminbnd(@(x)integral(@(t2)integral(@(t1)Int_fgf(xi,x,t1,
  t2, lambda), th(1,1), th(1,2), 'ArrayValued', true), th(2,1), th(2,2),
  'ArrayValued', true), X(1), X(2));
 rozdiel=abs(I_c-I_fqf);
 En_krit_pom=integral(@(t2)integral(@(t1)Entr_krit(xi,t1,t2,lambda),
 th(1,1),th(1,2),'ArrayValued',true),th(2,1),th(2,2),'ArrayValued',true);
 En_krit=(1/lambda) *log(En_krit_pom);
end
function [I]=Entr_krit(xi,t1,t2,lambda)
%pomocna funkcia na vypocet Entr. kriteria
 size_xi=size(xi);
 C=0;
 M=0;
      for k=1:size_xi(2)
          M=M+xi(2,k)*f_Theta(t1,t2,xi,k)*f_Theta(t1,t2,xi,k)';
      end
 Krit=kriterium(M);
  I=exp(lambda*Krit);
end
function [En_krit,Maximum,Max_Theta]=Minmax_optimalita(Theta,lambda,xi_e)
%funkcia na vypocet minimax kriteria
  size_xi_AVE=size(xi_e);
 Maximum=-Inf;
 Max_Theta=[];
  for t1=Theta(1,1):0.1:Theta(1,2)
      for t2=Theta(2,1):0.1:Theta(2,2)
          M_Theta=0;
          for k=1:size_xi_AVE(2)
              M_Theta=M_Theta+xi_e(2,k) *f_Theta(t1,t2,xi_e,k) *f_Theta(t1,
              t2,xi_e,k)';
          end
          Kriterium=kriterium(M_Theta);
          if (Kriterium==Maximum)
```

```
Maximum=Kriterium;
              Max_Theta=[Max_Theta;t1,t2];
          end
          if (Kriterium>Maximum)
              Maximum=Kriterium;
              Max_Theta=[t1, t2];
          end
      end
 end
 En_krit_pom=integral(@(t2)integral(@(t1)Entr_krit(xi_e,t1,t2,lambda),
 Theta(1,1), Theta(1,2), 'ArrayValued', true), Theta(2,1), Theta(2,2),
  'ArrayValued',true);
 En_krit=(1/lambda) *log(En_krit_pom);
end
function [ALPHA]=krok(alpha,t1,t2,xi,x_min_1,x_min_2,lambda)
%funkcia na vypocet optimalneho kroku
 size_xi=size(xi);
 M=0;
  for k=1:size_xi(2)
      M=M+xi(2,k)*f_Theta(t1,t2,xi,k)*f_Theta(t1,t2,xi,k)';
 end
 M_alpha=(1-alpha)*M+(alpha/2)*f_Theta_min(t1,t2,x_min_1)*f_Theta_min(t1,
 t2,x_min_1)'+ (alpha/2)*f_Theta_min(t1,t2,x_min_2)*f_Theta_min(t1,
 t2,x_min_2)';
 Krit=kriterium(M_alpha);
 ALPHA=exp(lambda*Krit);
end
function [xi_e,XMIN,KROK,ENKRIT,epsilon,m,cas]=Entropia_algoritmus(X,th,
lambda,eps,vaha)
%funkcia Entropijneho algoritmu
 tic
 xi = [X(1), X(1)/2 + X(2)/2, X(2); 1/3, 1/3, 1/3];
  [x_min_1, I_fgf]=fminbnd(@(x) integral(@(t2) integral(@(t1) Int_fgf(xi, x, t1,
 t2, lambda), th(1,1), th(1,2), 'ArrayValued', true), th(2,1), th(2,2),
  'ArrayValued',true),X(1),X(2));
 distance=x_min_1-X(1);
  x_min_2=X(2)-distance;
  XMIN=[];
```

```
XMIN=[XMIN, x_min_1, x_min_2]
[alpha, value]=fminbnd(@(alpha)integral(@(t2)integral(@(t1)krok(alpha,
t1,t2,xi,x_min_1,x_min_2,lambda),th(1,1),th(1,2),'ArrayValued',true),
th(2,1),th(2,2),'ArrayValued',true),0,1);
KROK = [];
KROK=[KROK, alpha/2, alpha/2];
xi=[[x_min_1;alpha/2], [X(1)/2+X(2)/2; (1-alpha)], [x_min_2;alpha/2]];
I_c=integral(@(t2)integral(@(t1)Int_c(xi,t1,t2,lambda),th(1,1),th(1,2),
'ArrayValued', true), th(2,1), th(2,2), 'ArrayValued', true);
I_c;
IC=[];
IC=[IC, I_c];
En_krit_pom=integral(@(t2)integral(@(t1)Entr_krit(xi,t1,t2,lambda),
th(1,1),th(1,2),'ArrayValued',true),th(2,1),th(2,2),'ArrayValued',true);
En_krit=(1/lambda) *log(En_krit_pom);
ENKRIT=[];
ENKRIT=[ENKRIT,En_krit]
m=1;
k=1;
epsilon=[];
epsilon=[epsilon, abs(I_c-I_fgf)/En_krit_pom]
while (abs(I_c-I_fgf)/En_krit_pom>eps)
    m = m + 1;
    size_xi=size(xi);
    n=size_xi(2);
    [alpha, value]=fminbnd(@(alpha)integral(@(t2)integral(@(t1)krok
    (alpha,t1,t2,xi,x_min_1,x_min_2,lambda),th(1,1),th(1,2),
    'ArrayValued', true), th(2,1), th(2,2), 'ArrayValued', true), 0,1);
    distance=x_min_1-X(1);
    x_min_2=X(2)-distance;
    XMIN=[XMIN, x min 1, x min 2];
    KROK=[KROK, alpha/2, alpha/2];
    xi=[xi(1,:);(1-alpha)*xi(2,:)];
    xi=[xi, [x_min_1; alpha/2], [x_min_2; alpha/2]];
    for i=1:n+1
        for j=1:n+1
            if (i \neq j \& abs(xi(1,i)-xi(1,j)) \leq vaha)
                 xi(1,i) = (xi(1,i) + xi(1,j)) / (2);
```

```
xi(2,i) = xi(2,i) + xi(2,j);
                xi(:,[j])=[];
                xi=[xi, [Inf; Inf]];
            end
        end
    end
    a=[];
    for i=1:n+1
        if (xi(1,i) \neq Inf)
            a=[a,xi(:,i)];
        end
    end
    b=sum(a(2,:));
    a=[a(1,:);(1/b)*a(2,:)];
    xi=a;
    [x_min_1, I_fqf]=fminbnd(@(x)integral(@(t2)integral(@(t1)Int_fqf(xi,
    x,t1,t2,lambda),th(1,1),th(1,2),'ArrayValued',true),th(2,1),th(2,2),
    'ArrayValued',true),X(1),X(2));
    I_c=integral(@(t2)integral(@(t1)Int_c(xi,t1,t2,lambda),th(1,1),
    th(1,2), 'ArrayValued', true), th(2,1), th(2,2), 'ArrayValued', true);
    En_krit_pom=integral(@(t2)integral(@(t1)Entr_krit(xi,t1,t2,lambda),
    th(1,1),th(1,2),'ArrayValued',true),th(2,1),th(2,2),
    'ArrayValued',true);
    En_krit=(1/lambda) *log(En_krit_pom);
    epsilon=[epsilon, abs(I_c-I_fgf)/En_krit_pom];
    ENKRIT=[ENKRIT, En_krit];
    xi_e=xi;
    abs(I_c-I_fgf)/En_krit_pom;
end
size_XMIN=size(XMIN);
XMIN=[XMIN(1:2), XMIN(5:(size_XMIN(1,2)))];
size_KROK=size(KROK);
KROK=[KROK(1:2), KROK(5:(size_KROK(1,2)))];
cas=toc;
```

```
end
```