Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Plzeň, 2017

Martin Černý

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím s odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 19. května 2017 . . . . . .

vlastnoruční podpis

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce, Ing. Zdeňkovi Hanzlíčkovi, Ph.D., za podnětné rady a připomínky při zpracování této práce.

Anotace

Tato práce se zabývá určením optimálního počtu stavů skrytých semi-Markovských modelů využívaných při parametrické statistické syntéze řeči. V první části jsou představeny některé přístupy k syntéze řeči a je vysvětlena struktura skrytých semi-Markovských modelů. V druhé části je uveden popis tří metod použitých pro získání optimální počtu stavů: analýza řečového signálu, shlukování na základě podobnosti stavů a shlukování stavů s nástroji HTK/HTS. Na závěr je proveden poslechový test porovnávající systém s navrženým počtem stavů a systém využívající standardní počet stavů.

Klíčová slova: parametrická statistická syntéza řeči, HTK/HTS, český jazyk, HSMM

Abstract

This thesis focuses on determining the optimal number of states of hidden semi-Markov models which are used in statistical parametric speech synthesis. First part describes basic approaches to speech syntesis and the structure of hidden semi-Markov models is explained. Second part presents three methods used for detemining the optimal number of states: analysis of the speech signal, clustering based on states similarity and clustering with HTK/HTS tools. At the end of the thesis the listening test, comparing the origin system and the designed system, is done.

Key words: statistical parametric speech synthesis, HTK/HTS czech language, HSMM

Obsah

1 Úvod

Aplikace řečové syntézy jsou v dnešním světě poměrně hojně rozšířeny. Ať už se jedná o pomocné nástroje a systémy pro zrakově či řečově postižené jedince, nebo v aplikacích, kdy počítač komunikuje s člověkem pomocí syntetizované řeči. Proces syntézy řeči prošel vývojem od systémů, kde výsledná řeč byla v podstatě jen hláskování jednotlivých slov, až do dnešní podoby, kdy vysyntetizovaná řeč je v mnoha případech téměř nerozeznatelná od přirozené řeči. Výzkum v této oblasti se dnes soustředí tedy především na to, aby syntéza zněla co nejméně uměle[1]. Dá se předpokládat, že v budoucnu bude syntéza řeči ještě víc využívána, s ohledem na potenciální aplikace tohoto odvětví a tendenci vše automatizovat. Systém syntézy řeči je většinou součástí systému TTS (Text To Speech). Úkolem TTS systému je ze vstupní textové informace vytvořit danou promluvu. Vlastní syntéze řeči předchází systém, který ze vstupního textu získá posloupnost hlásek a prozodickou informaci.

2 Syntéza řeči

Syntéza řeči je proces vytváření umělé lidské řeči. Systém, který řeč syntetizuje je syntetizér řeči. Na vstup se přivádí informace o požadované promluvě, tzn. její textová podoba, intonace, rytmus, atd. Existuje několik přístupů k syntéze.

2.1 Artikulační syntéza

Dnes asi nejméně používanou technikou je artikulační syntéza. Tato metoda se snaží modelovat celé hlasové ústrojí, tzn. hlasivky a mluvidla, a budící signál. Parametry tohoto modelu jsou anatomické parametry mluvidel a hlasivek a tlak plic. Tato metoda představuje intuitivní přístup k řešení problému, v dnešní době nedosahuje ovšem takových výsledků jako jiné metody[2].

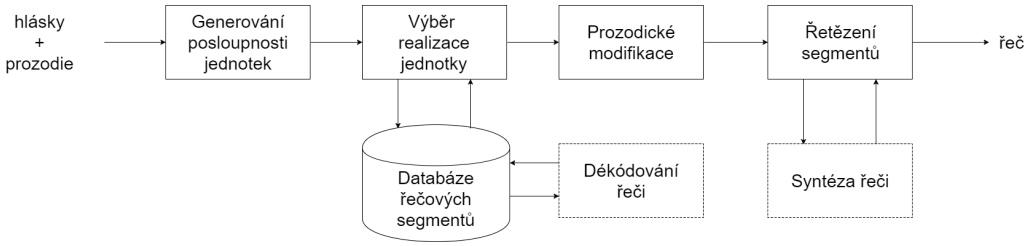
2.2 Formantová syntéza

Další metodou je formantová syntéza. Formantová syntéza modeluje také zdroj buzení a hlasové ústrojí, které je reprezentováno lineárním filtrem. Zdroj buzení generuje periodický signál, který se používá pro modelování znělých hlásek, a bílý šum, pro hlásky neznělé. Podle požadovaného výstupu se používá buď periodický zdroj, zdroj šumu, nebo jejich kombinace. Lineární filtr, reprezentující hlasové ústrojí se skládá ze sériového nebo paralelního spojení rezonátorů a antirezonátorů, které se implementují pomocí pásmové propusti. Tyto rezonátory a antirezonátory reprezentují formanty a antiformanty. Formant je určitá frekvenční oblast, která se při průchodu hlasového ústrojí zesiluje, naopak antiformant je frekvenční oblast která se zeslabuje.
K syntetyzování řeči se využívá pravidel. Tyto pravidla vznikají analýzou řečového korpusu, následnou parametrizací a uložením do databáze pravidel. Konkrétní hodnoty parametrů pravidel se poté nastavují v závislosti na konkrétním řečníkovi. Mezi hlavní výhody této metody patří malý počet parametrů, konstantní kvalita syntézy a možnost změny hlasu při úpravě parametrů (mužský hlas na ženský, šepot). Mezi hlavní nevýhody patří manuální, pracné určování pravidel[2].

2.3 Konkatenační syntéza

Dnes nejrozšířenější metodou je konkatenační syntéza. Tato metoda nemodeluje proces vytváření řeči, pracuje přímo s přirozeným řečovým signálem. Jak

název napovídá, tato syntéza funguje na principu řetězení určitých řečových jednotek. Při řetězení může docházet k nespojitostem a tudíž ke zřetelným chybám ve vysyntetizované promluvě.
Vytváření řečových jednotek může probíhat manuálně, kdy nějaký expert v této oblasti ručně vytvoří inventář řečových jednotek. Druhým způsobem je automatické vytvoření inventáře řečových jednotek, ke kterému je potřeba rozsáhlého řečového korpusu. Jak řečník, tak texty promluv musí být pečlivě vybrány. Promluvy by měl nahrávat profesionální řečník, který je schopen udržet delší dobu stejný styl mluvení a jehož hlas je příjemný na poslech. Věty řečového korpusu by měly být sestaveny tak, aby se v nich vyskytovaly všechny hlásky, kontexty a jiné řečové charakteristiky.
Dalším dělením této metody může být způsob řetězení jednotlivých jednotek. Jsou systémy které řetězí přímo, bez modifikace, řečové jednotky obsažené v inventáři. Tato metoda vyžaduje rozsáhlý inventář jednotek, tedy mnoho výskytů jednotlivých monofonů, tak aby se vždy vybral ten kontext, který je optimální pro danou situaci. Druhým způsobem je řetězení s modifikací. Tato metoda nevyžaduje tak rozsáhlý inventář a případné chyby při řetězení se snaží minimalizovat modifikací jednotek. Konkatenační metodu můžeme dále rozdělit podle velikosti korpusu nebo podle reprezentace řečových jednotek.
Proces syntézy řeči touto metodou je znázorněn na diagramu 1.



Obrázek 1: Schéma syntézy řeči konkatenační metodou

Těmito kroky se vytvoří inventář řečových segmentů potřebný pro syntézu.

1. Nejdříve je nutné rozhodnout jaké jednotky se budou používat (difony, trifony), jestli budeme řečový inventář vytvářet ručně nebo automaticky a také jakou strukturu bude mít řečový korpus.

2. Dále je potřeba segmentovat řečový korpus podle zvolených řečových jednotek. Segmentace může být buď manuální nebo automatická.

3. Poté můžeme pro každou jednotku z inventáře vybrat zástupce této jednotky, který se bude používat pro syntézu.

Další kroky popisují postup syntézy zadané promluvy.

1. Na vstup se přivede informace o požadované promluvě.

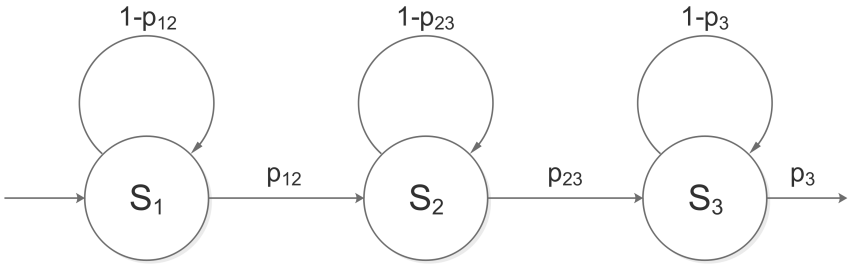
2. Z této informace se vytvoří posloupnost řečových jednotek, se kterými bude pracovat. Pro každou jednotku se vybere nejlepší reprezentace (pokud je jich více) z inventáře řečových segmentů. Pokud jsou řečové segmenty zakódovány, dochází k jejich dekódování.

3. Segmenty se řetězí, případně se vyhlazují akustické přechody mezi nimi a vzniká výsledná řeč.

Mezi výhody této metody patří zejména to, že nemodeluje proces vytváření řeči člověkem, pracuje přímo s řečovým signálem. Výsledná řeč je také obecně mnohem kvalitnější než u formantové syntézy. Při špatné volbě řečového korpusu naopak může dojít k výraznému poklesu kvality, pokud se požadovaná jednotka nenachází v inventáři. To, že musíme uchovávat v paměti všechny velké množství řečových segmentů, vede k paměťově vyšší náročnosti než u formantové metody.

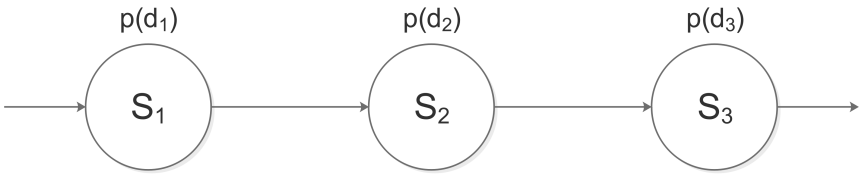
2.4 Parametrická statistická syntéza řeči

V systému se, kterým jsem pracoval, je použita metoda parametrické statistické syntézy řeči. Jedná se o konkatenační metodu, která nepracuje přímo s řečovými segmenty, ale využívá pro reprezentaci jednotlivých řečových jednotek skrytých semi-Markovských modelů (hidden semi-Markov model, HSMM).
Nejprve si vysvětlíme co jsou to Markovské modely. Markovské modely jsou klasické Markovské řetězce. Máme tedy n počet stavů a pravděpodobnosti přechodu mezi nimi. Vlastnost skryté modely znamená, že jednotlivé stavy jsou nepozorovatelné, tzn. že nevíme přesně, kdy jsme ve kterém stavu, známe pouze výstup. Modely s těmito s vlastnostmi nazývají skryté Markovské modely (HMM). Model se třemi stavy je znázorněn na obr. 2



Obrázek 2: Schéma HMM

Pro účely syntézy řeči tyto modely nejsou vhodné, neboť konstantní hodnoty pravděpodobností přechodu nepopisují trvání stavu přirozeným ani praktickým způsobem. Proto se používají modely HSMM, semi-Markovské modely. Tyto modely se liší od klasických tím, že neobsahují pravděpodobnosti přechodu, ale každý stav má svoji pravděpodobnost trvání. Tato pravděpodobnost je dána normálním rozdělením. Třístavový model je znázorněn na obr. 3



Obrázek 3: Schéma HSMM

Samotný proces syntézy obsahuje dvě části. První část zahrnuje trénování HSMM. Parametrizací promluv z řečového korpusu získáme parametrické vektory. Tyto vektory poté reprezentujeme pomocí HSMM. Trénování modelů probíhá nejprve na úrovni monofonů. Poté se přechází na kontextově závislé jednotky, tzn. jednotky, u kterých je brán v potaz jejich pravý a levý kontext, pozice ve větě, atd. Nakonec dochází ke shlukování modelů pomocí rozhodovacích stromů.
Druhá část obsahuje již syntézu požadované promluvy. Ze vstupní informace se vytvoří posloupnost požadovaných řečových jednotek. Pomocí rozhodovacích stromů, obsahující např. otázky na levý a pravý kontext, se nalezne odpovídající model. Pomocí speciálního algoritmu se vygeneruje posloupnost parametrických vektorů, ze kterých se zrekonstruuje výsledná řeč[3][4].
V práci jsem pracoval se dvěma hlasy, mužským řečníkem *AJ* a ženským řečníkem *KI*. Pro oba dva jsem se snažil odhadnout optimální počet stavů HSMM modelů. Protože systémy fungující na principu parametrické syntézy řeči většinou používají stavů 5 pro všechny hlásky, rozhodl jsem se omezit možný počet. Zdola jsem zvolil omezení na počet stavů 2, protože model s jedním stavem by už prakticky neměl význam. Horní omezení jsem zvolil na 7 stavů, větší počet by vedl k přetrénování.

2.5 Fonetická abeceda

V mezinárodním měřítku se pro záznam a práci s hláskami používá Mezinárodní fonetická abeceda IPA (International phonetic alphabet). IPA používá pro popis hlásek kromě klasické latinky také speciální fonetické znaky, známé například z fonetického přepisu anglických slov (apple['æpl]). Další rozšířenou fonetickou abecedou je SAMPA (Speech Assessment Methods Phonetic Alphabet), která, jak název napovídá, je určena pro lepší reprezentaci hlásek v počítači. Dále existuje i Česká fonetická abeceda, která reprezentuje

pouze české hlásky. V systému, se kterým jsem pracoval, se používá interní fonetická abeceda EPA, pomocí které jsou také hlásky v této práci označeny. Tato abeceda je tvořena pouze velkými a malými latinskými písmeny, a je tak vhodná pro snadnou práci a programování. Přehled jednotlivých abeced je znázorněn v tabulce 1, převzato z publikace[2]

	IPA	SAMPA	EPA	příklad		IPA	SAMPA	EPA	příklad
vokály	ɪ	i	i	*l**i**s*	plozivy	p	p	p	***p**ec*
	ɛ	e	e	*p**e**s*		b	b	b	***b**ratr*
	a	a	a	*s**a**d*		t	t	t	***t**uk*
	o	o	o	*k**o**v*		d	d	d	***d**ům*
	u	u	u	*s**u**kně*		c	c	T	*dě**t**i*
	i:	i:	I	*v**í**no*		ɟ	J\	D	***d**ítě*
	ɛ:	e:	E	*l**é**k*		k	k	k	***k**ost*
	a:	a:	A	*s**á**l*		g	g	g	*ty**g**r*
	o:	o:	O	*k**ó**d*	nazály	m	m	m	***m**už*
	u:	u:	U	*r**ů**že*		n	n	n	*ví**n**o*
diftongy	o͡u	o_u	y	*b**ou**da*		ɲ	J	J	*la**ň**ka*
	a͡u	a_u	Y	***au**to*	afrikáty	t͡s	t_s	c	***c**ena*
	e͡u	e_u	F	***eu**nuch*		t͡ʃ	t_S	C	*o**č**i*
frikativy	f	f	f	***f**ík*		d͡z	d_z	w	*pod**z**im*
	v	v	v	***v**ítr*		d͡ʒ	d_Z	W	*dž**b**án*
	s	s	s	***s**ůl*	významné alofony	ŋ	N	N	*ta**n**go*
	z	z	z	*ko**z**a*		ŋ̍	F	M	*tra**m**vaj*
	ʃ	S	S	***š**kola*		ɣ	G	G	*aby**ch** byl*
	ʒ	Z	Z	***ž**ena*		r̝̊	Q\	Q	*t**ř**i*
	x	x	x	***ch**ata*		r̥	r=	P	*k**r**k*
	ɦ	h\	h	***h**ůl*		l̥	l=	L	*vl**k*
	l	l	l	*vl**a**k*		m̥	m=	H	*os**m***
	r	r	r	***r**ok*		ʔ	?	!	*"ráz"*
	r̝	P\	R	*mo**ř**e*		ə	@	@	*"šva"*
	j	j	j	***j**ev*					

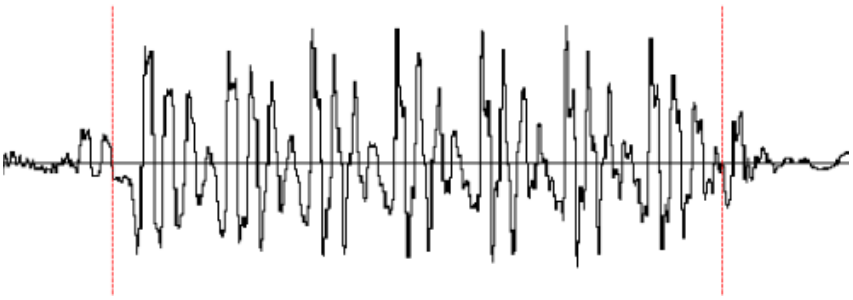
Tabulka 1: Přehled fonetických abeced

3 Zkoumání řečového signálu v časové oblasti

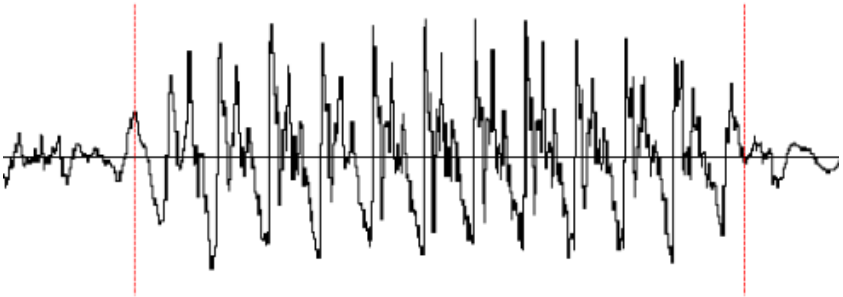
V této části jsem se pokusil odhadnout optimální počet stavů HSMM pomocí analýzy průběhu řečového signálu v časové oblasti. K dispozici jsem měl několik promluv od obou řečníků. Pro každou hlásku jsem v daných promluvách našel několik výskytů v různých kontextech, tzn. na začátku slova, uprostřed i na konci. Poté jsem zkoumal časový průběh těchto výskytů (viz obr. 4) a snažil se odhadnout počet stavů. K dispozici jsem měl také segmetaci pro jednotlivé promluvy, která jednotlivé promluvy rozdělí na úseky hlásek. Tato segmentace je vytvořená automaticky, nemusí být proto přesná, neboť správná automatická klasifikace představuje další rozsáhlý problém. Ke zkoumaní řečového signálu jsem využil zejména software Wavesurfer a Audacity. Tato metoda není nijak složitá a není podložena žádným výpočtem. Jedná se spíše o intuitivní přístup k problému a první odhad řešení.

Obrázek 4: Ukázka časového průběhu signálu

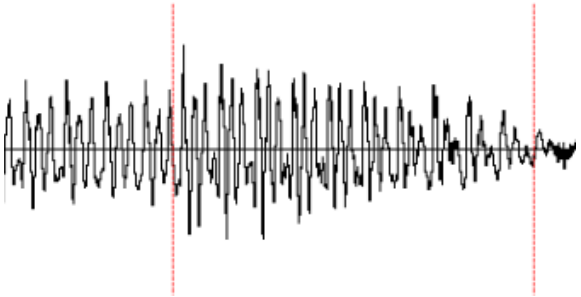
Pokud se podíváme na jednotlivé skupiny hlásek, všimneme si, že mají podobný průběh signálu. Pro ukázku průběhu samohlásek jsem vybral monofon *e*. Pro monofon *e* jsem pro ukázku vybral slova *terčem* (první výskyt) a *kole*. Pro každé slovo jsem porovnal průběh mužského a ženského řečníka. Červeně jsou označeny přibližné hranice monofonu v signálu.



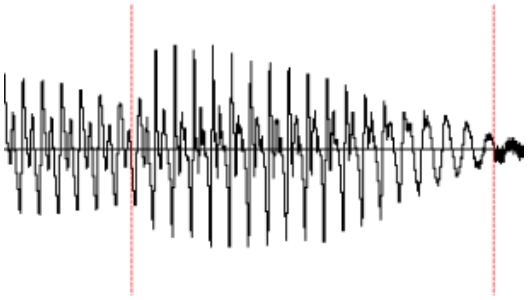
Obrázek 5: Průběh signálu pro monofon *e* ve slově *terčem* pro řečníka AJ



Obrázek 6: Průběh signálu pro monofon *e* ve slově *terčem* pro řečníka KI

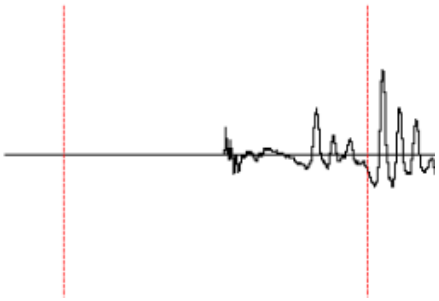


Obrázek 7: Průběh signálu pro monofon *e* ve slově *kole* pro řečníka AJ

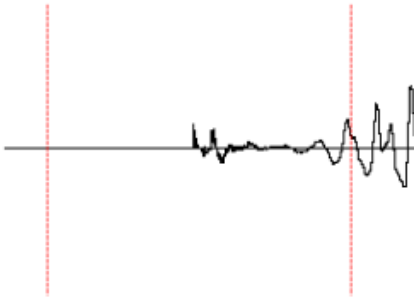


Obrázek 8: Průběh signálu pro monofon *e* ve slově *kole* pro řečníka KI

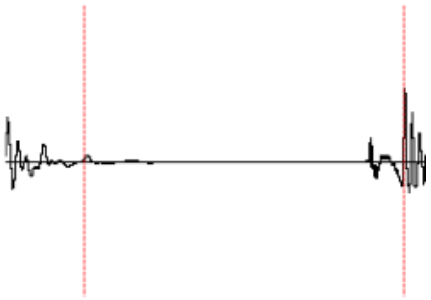
Z průběhu signálu jsem zvolil počet stavů tři. Průběh monofonu se zdá téměř konstantní po celou jeho dobu, tzn. že jeden stav máme pro průběh monofonu a dva krajní stavy představují koartikulační vztahy mezi sousedícími monofony. Takovýto průběh se opakuje u všech samohlásek, tzn. pro všechny samohlásky jsem zvolil počet stavů na tři.
Další zajímavou skupinou hlásek jsou explozivy. Tato skupina má svůj charakteristický průběh, který demonstruje monofon *p*. Pro ukázku jsem vybral slova *pozorný* a *vypadl*.



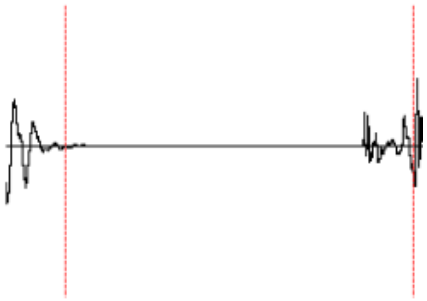
Obrázek 9: Průběh signálu pro monofon *p* ve slově *pozorný* pro řečníka AJ



Obrázek 10: Průběh signálu pro monofon *p* ve slově *pozorný* pro řečníka KI



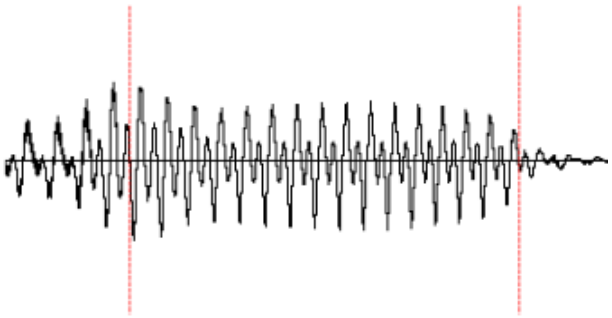
Obrázek 11: Průběh signálu pro monofon *p* ve slově *vypadl* pro řečníka AJ



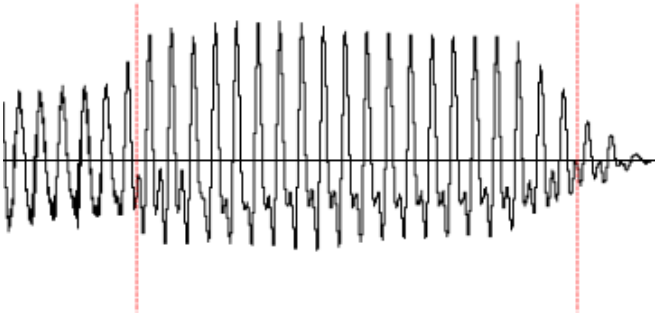
Obrázek 12: Průběh signálu pro monofon *p* ve slově *vypadl* pro řečníka KI

Na signálu je patrná věc společná pro všechny plozivy. Signál zůstává dlouhou dobu na nule a poté dojde k explozi. Proto jsem pro většinu ploziv zvolil počet stavů čtyři.
Tímto způsobem jsem provedl analýzu pro všechny hlásky a určil jejich počet stavů. Pro hlásky pro které jsem neměl dostatek výskytů a tedy nemohl jsem

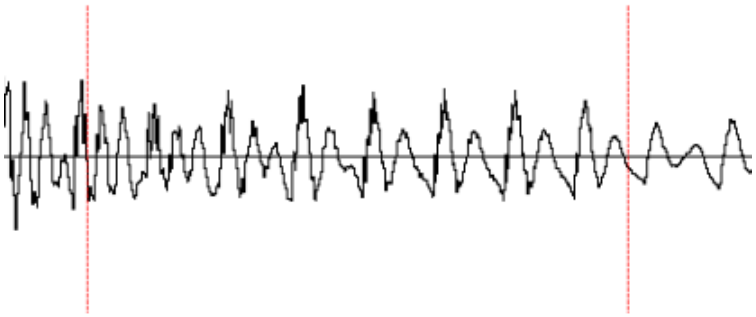
provést analýzu signálu (G, M) jsem počet stavů zvolil stejný jako pro jejich neznělé ekvivalenty. Mezi výsledky jsou zajímavé především hlásky *m* a *j*, pro které jsem určil optimální počet stavů na dva stavy. Podle časového průběhu se totiž zdá, že samy o sobě nemají nijak výrazný průběh a přejímají pouze průběh z kontextu. Pro monofon *m* uvádím průběh slova *tímto*, pro monofon *j* uvádím průběh slova *zájmu*.



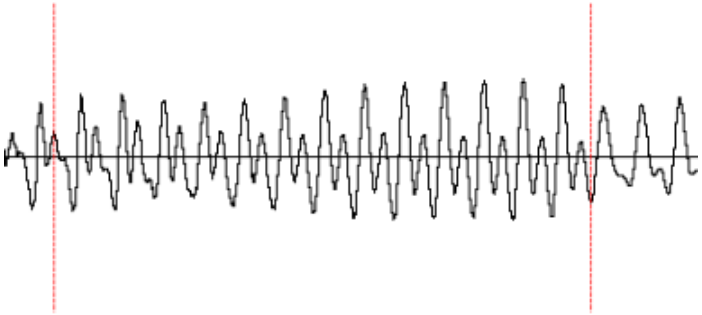
Obrázek 13: Průběh signálu pro monofon *m* ve slově *tímto* pro řečníka AJ



Obrázek 14: Průběh signálu pro monofon *m* ve slově *tímto* pro řečníka KI



Obrázek 15: Průběh signálu pro monofon *j* ve slově *zájmu* pro řečníka AJ



Obrázek 16: Průběh signálu pro monofon *j* ve slově *zájmu* pro řečníka KI

Na závěr metody uvádím tabulku výsledků pro všechny hlásky (viz tabulka 2).

Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů
i	3	O	3	z	3	j	2	g	4	N	4
e	3	U	3	S	4	p	4	m	2	M	2
a	3	y	4	Z	3	b	3	n	3	G	3
o	3	Y	4	x	3	t	4	J	3	Q	3
u	3	F	4	h	3	d	4	c	4	P	5
I	3	f	4	l	3	T	5	C	4	L	3
E	3	v	3	r	4	D	5	w	4	H	3
A	3	s	4	R	3	k	4	W	3		

Tabulka 2: Výsledné počty stavu pro metodu analýzy řečového signálu

4 Metoda podobnosti stavů na základně KL-divergence

Jako další metodu jsem použil shlukování stavů na základě jejich podobnosti. Myšlenka je taková, že pokud se pravděpodobnostní rozdělení dvou sousedících stavů málo liší, můžeme přejít k modelu, který má počet stavů o jeden menší, tzn. k modelu jednoduššímu. Jako míru podobnosti jsem použil Kullback - Leiblerovu divergenci (KL divergenci). Pro výpočet a znázornění KL divergence mezi jednotlivými stavy jsem použil skript v Pythonu. Vstupem programu byly textové soubory s definicí Markových modelů a výstupem byly KL divergence mezi jednotlivými stavy.

4.1 KL divergence

Jako míru pro porovnání stavů jsem použil KL divergenci. Tato míra určuje podobnost dvou pravděpodobnostních rozložení. KL divergence pro dvě vícerozměrné normální pravděpodobnostní rozložení má následující vzorec[5]:

$$D_{KL}(N_0||N_1) = \frac{1}{2}\left(tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) - k + ln\left(\frac{det\Sigma_1}{det\Sigma_0}\right)\right),$$

kde μ_1, μ_0 představují vektory středních hodnot, Σ_1, Σ_0 jsou kovarianční matice, v mém případě diagonální matice která má na diagonále právě vektor rozptylů. Číslo k udává dimenzi parametrického vektoru. Funkce $tr(x)$, udává stopu matice, tzn. součet prvků na hlavní diagonále.

4.2 Vstupní soubory

```
~o
<STREAMINFO> 5 120 1 1 1 63
<MSDINFO> 5 0 1 1 1 0
<VECSIZE> 186<NULLD><USER><DIAGC>
~h "a"
.
.
.
<BEGINHMM>
<NUMSTATES> 4
<STATE> 2
<SWEIGHTS> 5
 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 ...
<STREAM> 1
<MEAN> 120
 -3.378209e+00 2.385735e+00 1.369131e-01 -1.317788e-02 ...
<VARIANCE> 120
 1.243047e-01 7.303812e-02 1.190445e-01 5.115412e-02 ...
<GCONST> -4.433881e+02
<STREAM> 2
.
.
.
<TRANSP> 4
 0.000000e+00 1.000000e+00 0.000000e+00 0.000000e+00
 0.000000e+00 9.219311e-01 7.806886e-02 0.000000e+00
 0.000000e+00 0.000000e+00 6.738284e-01 3.261716e-01.
 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
<ENDHMM>
```

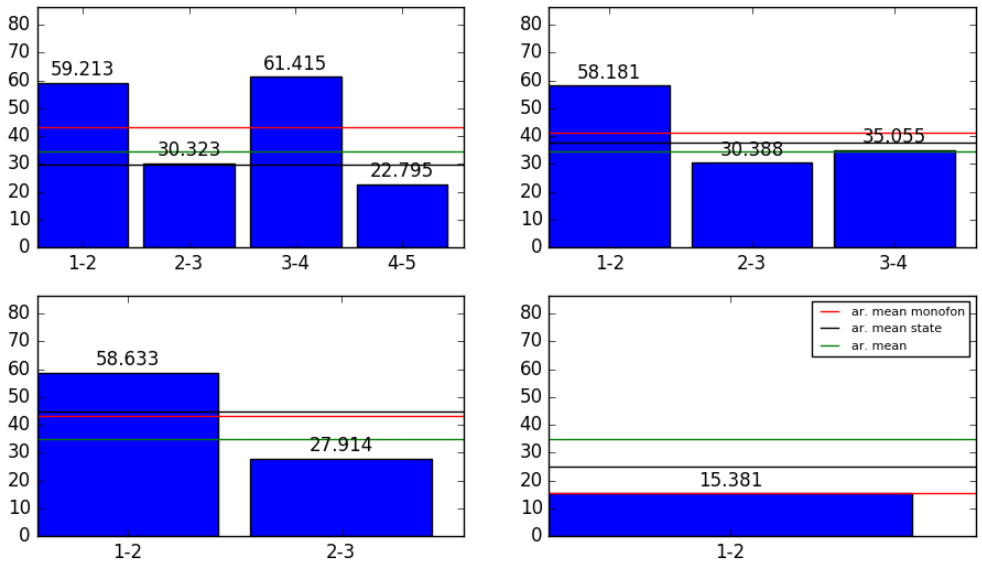
Vstupními soubory, jsou textové soubory, obsahující definici modelu s určitým počtem stavů. Hlavička souboru specifikuje počet a dimenzi jednotlivých proudů parametrů a určuje o jakou hlásku se jedná. Další část souboru je tělo, kde jsou uloženy informace o jednotlivých stavech. V práci se zabývám pouze prvním proudem parametrů obsahujícím kepstrální koeficienty. Další proudy obsahující základní hlasivkovou frekvenci a koeficienty aperiodicity se pro jednoduchost ignorují. Každý stav je tedy složen ze tří proudů parametrů, přičemž ten se kterým pracuji má dimenzi 120. V tomto vektoru jsou ulo-

žený jak statické parametry, tak i dynamické parametry prvního a druhého stupně. V souboru je uveden počet stavů vždy o dva víc, a to proto, že model obsahuje i jeden počáteční a jeden koncový stav navíc, kvůli přechodu mezi hláskami. Dále už jsou uvedeny vektory středních hodnot a vektor variance, tedy hodnoty, se kterými pracuji. Na konci souboru je uvedena matice přechodů mezi jednotlivými stavy.

4.3 Postup řešení

Nejprve jsem napsal funkci, která počítá KL divergenci mezi dvěma pravděpodobnostními rozloženími. Vstupem této funkce jsou vektory středních hodnot a vektory variancí. Pro účely mé práce jsem použil hodnoty středních hodnot a variancí pouze z prvního streamu, který má dimenzi 120. Pro určení optimálního počtu jsem se z počátku omezil na modely o počtu stavů 2,3,4 a 5, mezi kterými jsem vybíral ten optimální.
Pro vizualizaci výsledných hodnot jsem využil sloupcového grafu, z něhož jsou hodnoty dobře interpretovatelné. Na x-ové ose grafu jsou jednotlivé dvojice stavů a na ose y jsou hodnoty KL divergence. Grafy pro všechny čtyři stavy jsem uvedl do jednoho grafu. Osy y jsou pro všechny stavy stejné, tak aby byly dobře vzájemně porovnatelné. V grafech jsou znázorněny tři hodnoty:

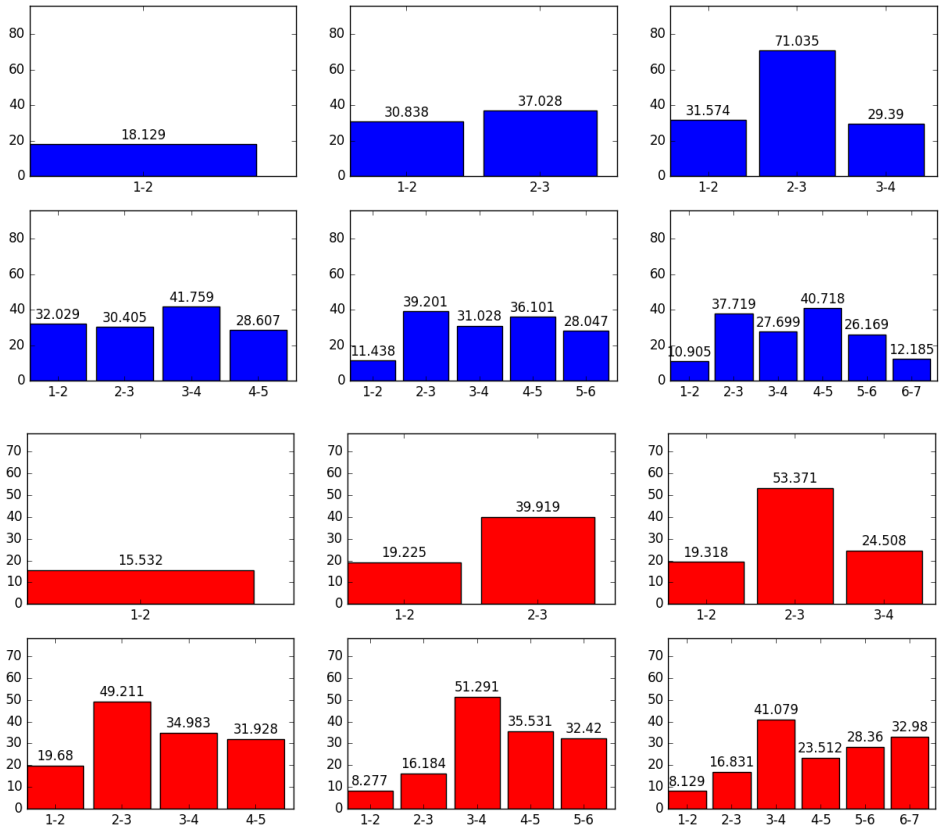
- zelená křivka - celková průměrná hodnota KL divergence
- černá křivka - průměrná hodnota KL divergence pro jednotlivé stavy, tzn. průměrná hodnota pro modely se dvěma, třemi, čtyrmi pěti stavy
- červená křivka - průměrná hodnota KL divergence v rámci jednoho modelu a jednoho stavu



Obrázek 17: Výsledný graf hlásky *b* pro řečníka AJ

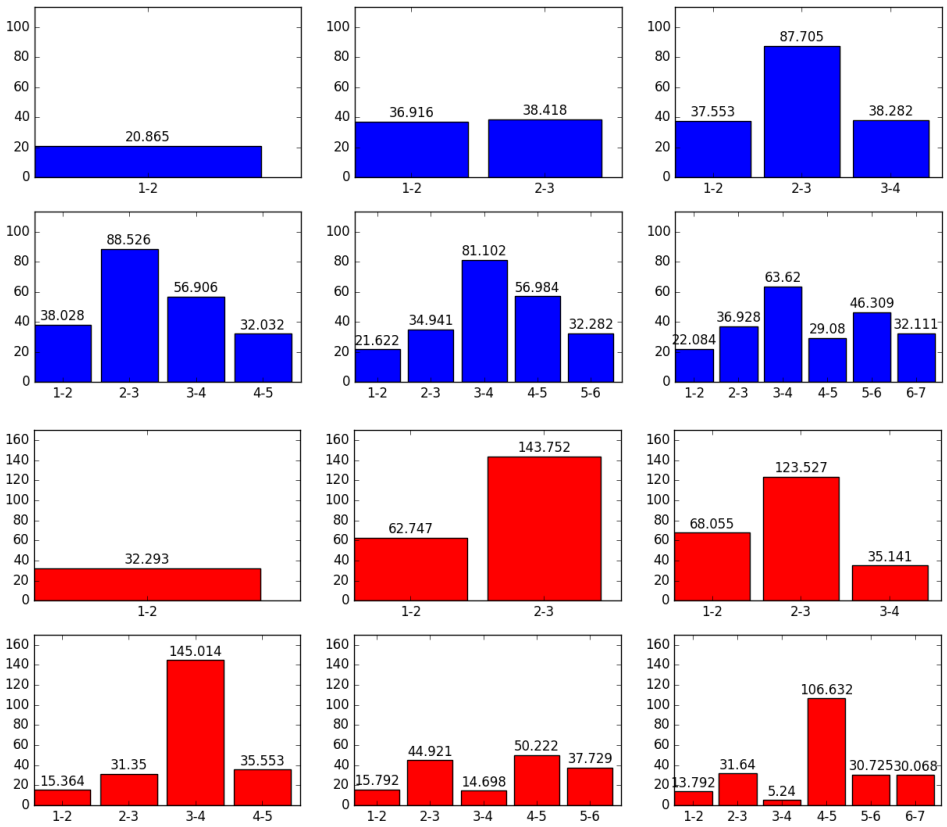
Nyní je potřeba určit optimální počet parametrů, což je asi ta nejtěžší část. Podle výše uvedených hodnot průměrů a hodnot KL divergence jsem navrhl jednoduché pravidlo určování počtu stavů. Při analýze výsledků jsem ale zjistil že toto pravidlo ne vždy funguje. Navíc hodnoty KL divergence jsou pro oba řečníky různé, a dá se tedy předpokládat, že obecně budou různé pro různé řečníky. Rozhodl jsem se tedy ustoupit od rozhodovacího pravidla a určit počet stavů ručně. V dalším kroku jsem se také rozhodl rozšířit maximální počet stavů na 7 a přejít od klasické KL divergenci k symetrické KL divergenci.

$$D_{KLS}(N_0||N_1) = \frac{1}{2}\Big(D_{KL}(N_0||N_1) + D_{KL}(N_1||N_0)\Big)$$



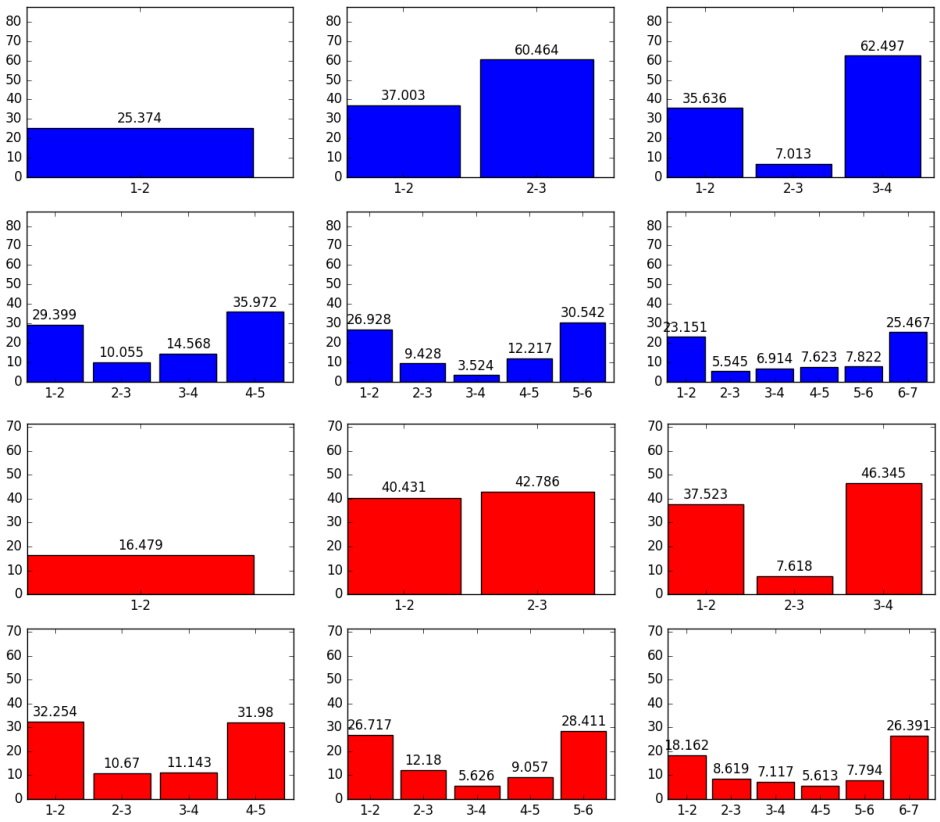
Obrázek 18: Hodnoty KL divergence pro monofon g

Na grafech můžeme vidět podobný trend pro oba řečníky, přestože se jednotlivé hodnoty mírně liší. Proces rozhodování probíhal od největšího počtu stavů po nejmenší. Nejvýraznější změny vidíme mezi modely se stavy pět a čtyři. Vysoká hodnota mezi stavy dva a tři u čtyřstavového modelu znamenají, že pokud bychom zvolili počet stavů čtyři, ztratili bychom přesnost. Pro monofon g jsem zvolil počet stavů 5.



Obrázek 19: Hodnoty KL divergence pro monofon *b*

Pro monofon *b* jsem zvolil počet stavů 4. Podle průběhu hodnot je vidět že k výrazné změně dojde mezi počty stavů tři a čtyři. Výsledný počet stavů bude tedy tři nebo čtyři. Kvůli vysoké hodnotě mezi druhým a třetím stavem u čtyřstavového modelu zůstanu na počtu čtyři, neboť při zvolení počtu tři by mohlo dojít ke ztrátě kvality.



Obrázek 20: Hodnoty KL divergence pro monofon *e*

Na monofon *e* jsem použil stejný proces rozhodování. V tomto případě jsem se však zastavil na počtů stavů 3, neboť malá hodnota mezi stavy 2 a 3 u čtyřstavového modelu nám značí možnost přejít na model o stav nižší. Podobně jsem rozhodl u všech hlásek a výsledky zanesl do tabulky (viz tabulka 3). Grafy hodnot KL divergence pro ostatní monofony jsou obsaženy v příloze.

Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů
i	3	O	3	z	4	j	3	g	5	N	3
e	3	U	3	S	4	p	4	m	3	M	3
a	3	y	4	Z	3	b	4	n	4	G	4
o	3	Y	4	x	3	t	5	J	3	Q	3
u	3	F	4	h	3	d	4	c	3	P	5
I	3	f	4	l	3	T	5	C	3	L	3
E	3	v	3	r	4	D	5	w	4	H	4
A	3	s	4	R	4	k	4	W	4		

Tabulka 3: Výsledné počty stavu pro metodu KL divergence

5 Shlukování stavů pomocí HTK/HTS

V další metodě jsem na sousední stavy použil shlukovací algoritmus implementovaný v HTK/HTS. HTS (HMM-based Speech Synthesis System) vlastně představuje patch pro HTK. Tento nástroj umožňuje práci s HSMM[6]. Cílem shlukovacího algoritmu je sloučit do jedné skupiny stavy, které jsou si podobné. Pro shlukování máme dva základní přístupy:

- Divizní metoda - Na začátku algoritmu máme jeden shluk, který obsahuje všechny objekty. Postupně dochází k dělení shluků na menší, podle určitých kritérií. Zastavovací podmínka algoritmu může být např. dosažení požadovaného počtu shluků, kritérium podobnosti objektů ve shluku nebo vzdálenosti shluků.

- Aglomerativní metoda - Na začátku algoritmu každý objekt představuje jeden shluk. Dále dochází ke spojování jednotlivých shluků. Zastavovací podmínky algoritmu jsou stejné jako pro divizní metodu, tedy dosažení požadovaného počtu shluků, kritérium podobnosti objektů ve shluku nebo vzdálenosti shluků.

V mém případě jsem použil divizní metodu shlukování. Pomocí shlukových otázek dochází k dělení shluků. HTK/HTS ovšem neumožňuje shlukovat jednotlivé stavy vzájemně. Musíme tedy přejít k rozdělení na jednostavové modely, kde index stavu je součást názvu modelu. Obecný řetězec pro model, který je rozdělen do jednotlivých stavů má tvar:

$$\text{hláska}\hat{}\text{index stavu zepředu}\hat{}\text{index stavu zezadu}$$

Pro třístavový model monofonu a vypadají rozložené modely následovně:

$$(a\hat{}1\hat{}3, a\hat{}2\hat{}2, a\hat{}3\hat{}1)$$

Přičemž shlukování probíhá vždy pro všechny jednostavové modely pro jednotlivé hlásky zvlášť.

Shlukové otázky, jejichž příklady jsou uvedeny níže, slouží k rozdělení shluku. Například otázka *StateBw=5or6* rozdělí shluk na dva shluky, kdy modely, které mají index stavu zezadu je 5 nebo 6, představují jeden shluk a druhý shluk představují ostatní modely.

```
QS "StateBw=5or6"
{ *^5 , *^6 }

QS "StateFw=6or7"
{ *^6^* , *^7^* }

QS "StateBw=6or7"
{ *^6 , *^7 }

QS "StateFw<=2"
{ *^1^* , *^2^* }

QS "StateBw<=2"
{ *^1 , *^2 }

QS "StateFw<=3"
{ *^1^* , *^2^* , *^3^* }
```

Při shlukování beru v úvahu pouze kepstrální parametry a parametr základní hlasivkové frekvence. Koeficienty aperiodicity zanedbávám. Z výsledných počtů stavů pro oba parametry jsem vybral pro každou hlásku ten větší. Pokud bych zvolil nižší počet stavů, mohlo by to způsobit nepřesnosti a zhoršení výsledků.
Zastavovací podmínka algoritmu je závislá na hodnotě kritéria MDL (Minimum Description Length). Kritérium je nastaveno pomocí jednoho váhového parametru v konfiguračním souboru. Mým cílem bylo nastavit tento parametr tak, aby shlukovací algoritmus dával přijatelné výsledky. Při experimentálním nastavování této konstanty jsem však objevil, že pro hlásky, jejichž výskyt je v promluvách velmi řídký (typicky např. O, Y), se musí zvolit tato konstanta několikrát nižší než pro ostatní hlásky. Jde si lehce představit, že pokud mám pro jednu hlásku například 10 výskytů a pro druhou 1000, musí se lišit i zastavovací podmínka. Kdybych těchto 10 výskytů dělil tolikrát jako 1000 výskytů, na konci dostanu vždy shluky jenom s jedním objektem. Pro tyto méně časté hlásky mi tedy vycházel počet stavů 1. Zprvu jsem se tento pro-

blém snažil vyřešit omezením počtu stavů zdola. Jako dolní omezení jsem bral výsledky z metody analýzy řečového signálu. Ani tyto výsledky ovšem nebyli uspokojivé a tak jsem se rozhodl tento přístup zavrhnout. V dalším kroku jsem zkoušel měnit i jiné parametry, výsledek byl ovšem stejný.
Tato metoda je tedy pro daný problém nevhodná, z důvodu nerovnoměrného výskytu jednotlivých hlásek.

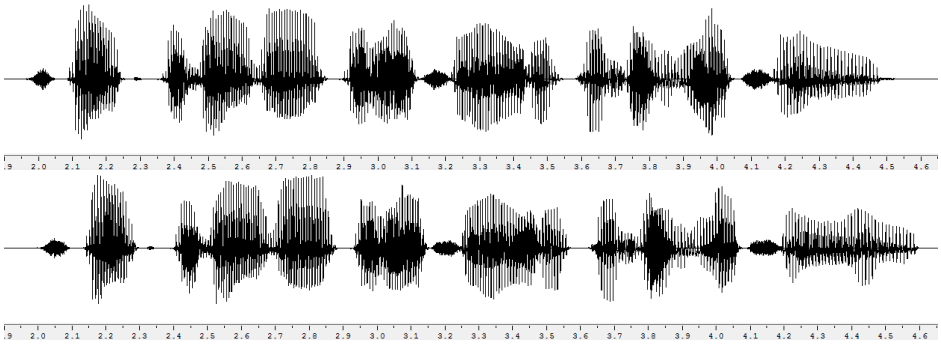
6 Poslechový text

Po aplikování všech metod jsem přistoupil k poslechovému testu. Pro poslechový test jsem se rozhodl zkombinovat výsledky prvních dvou metod a ty porovnat se standardním systémem, který využívá 5-ti stavové modely. Pro ty hlásky, pro které se výsledky metod lišily, jsem znovu zkoumal časový průběh signálu a hodnoty KL divergence mezi stavy. Větší váhu jsem přikládal metodě s KL-divergecí, neboť tato metoda je podložena výpočtem. Výsledný počet stavů jsem zanesl do tabulky (viz tabulka 4).

Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů	Monofon	Počet stavů
i	3	O	3	z	4	j	3	g	5	N	4
e	3	U	3	S	4	p	4	m	3	M	3
a	3	y	4	Z	3	b	4	n	3	G	3
o	3	Y	4	x	3	t	4	J	3	Q	3
u	3	F	4	h	3	d	4	c	4	P	5
I	3	f	4	l	3	T	5	C	4	L	3
E	3	v	3	r	4	D	5	w	4	H	4
A	3	s	4	R	4	k	4	W	4		

Tabulka 4: Počty stavů navrženého systému

Pro poslechový test jsem vybral 25 promluv od každého řečníka, které se nejvíce liší z hlediska průběhu signálu. Pro ilustraci přikládám ukázku časového průběhu té samé promluvy vysyntetizované původním a navrženým systémem. Z ukázek je vidět že amplitudy a trvání se mírně liší. Tyto rozdíly jsou ovšem ve výsledku jen mírně slyšitelné.

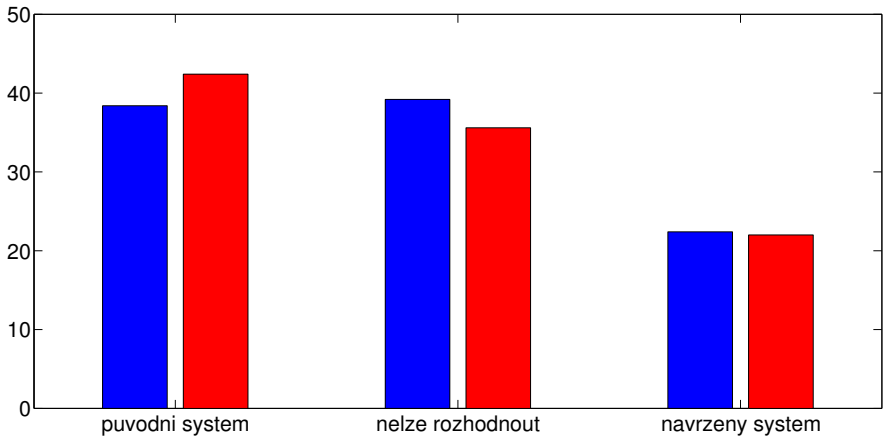


Obrázek 21: Výsledná promluva pro systém s pěti stavovými modely (horní obrázek) a výsledná promluva pro navržený systém (spodní obrázek)

Poslechový test vyplnilo 10 respondentů, z většiny se jednalo o studenty kybernetiky. Testovací otázka se skládá z dvou stejných promluv. Po poslechu obou promluv respondent zvolil promluvu, která mu zněla lépe, popřípadě zvolil možnost že nejde rozhodnout. Z výsledného počtu stavů můžeme ještě před testem odhadnout jeho možné výsledky. Průměrný počet stavů se značně snížil, nemůžeme tedy čekat zlepšení ve kvalitě výsledných promluv. Nejlepší možný výsledek by bylo zachování kvality syntézy při snížení počtu stavů, čímž by se snížili paměťová náročnost. Další možný výsledek je zhoršení kvality, což by znamenalo, že počty stavů navrženého systému jsou moc nízké. Výsledné odpovědi jsem zaznamenal do tabulky (viz tabulka 5) a také zakreslil do grafu (graf 22).

řečník	původní systém	nelze rozhodnout		navržený systém
		totožné znění	různé znění	
AJ	38.40%	19.20%	20.00%	22.40%
KI	42.40%	18.80%	16.80%	22.00%
průměr	40.40%	19.00%	18.40%	22.20%

Tabulka 5: Tabulka výsledků poslechového testu



Obrázek 22: Graf výsledků poslechového testu

Z vyhodnocení testu je vidět, že výsledky jsou konzistentní pro oba řečníky. Původní systém s pětistavovými modely vychází jako lepší, nicméně všichni respondenti se shodli na tom, že promluvy jsou si dost podobné a rozdíly mají pouze lokální charakter, tzn. navržený systém není výrazně horší než systém původní.

7 Závěr

Cílem této práce bylo navrhnout optimální počet stavů HSMM pro jednotlivé hlásky. V první části jsme zkoumali signály z řečového korpusu v časové oblasti. Touto metodou jsme dostali určitý odhad počtu stavů. V další metodě jsme použili podobnost mezi stavy, pro určení jejich optimálního počtu. Tato metoda nám dala výpočtem podložené výsledky. Jako poslední metodu jsme zvolili shlukování stavů za využití HTK/HTS. Tato metoda se ukázala pro tento typ problému nevhodná. Pro použití této metody by se hodil korpus, kde jsou jednotlivé hlásky rovnoměrně zastoupeny. Po aplikování metod jsme zvolili výsledný systém, který jsme porovnali se standardním pěti stavovým systémem. Průměrný počet stavů navrženého systému je výrazně nižší než pět, paměťová náročnost se tedy snížila. Z poslechového testu vyšel jako lepší původní systém. Jednotlivé promluvy se ale málo lišily, což potvrdila většina respondentů.
Zlepšení výsledků je možné hlubší analýzou průběhů KL divergence. Je také možné experimentovat s jinou míru pro určení podobnosti stavů jednotlivých modelů.

Seznam obrázků

Seznam tabulek

Reference

[1] Jindřich Matoušek. *Syntéza řeči*. Dostupné z: http://www.fit.vutbr.cz/study/courses/ZRE/public/pred/12_synteza_matousek/tts_tisk.pdf, 2015.

[2] Luděk Müller Vlasta Radová Josef Psutka, Jindřich Matoušek. *Mluvíme s počítačem česky*. Academia, 2006. ISBN: 80-200-1309-1.

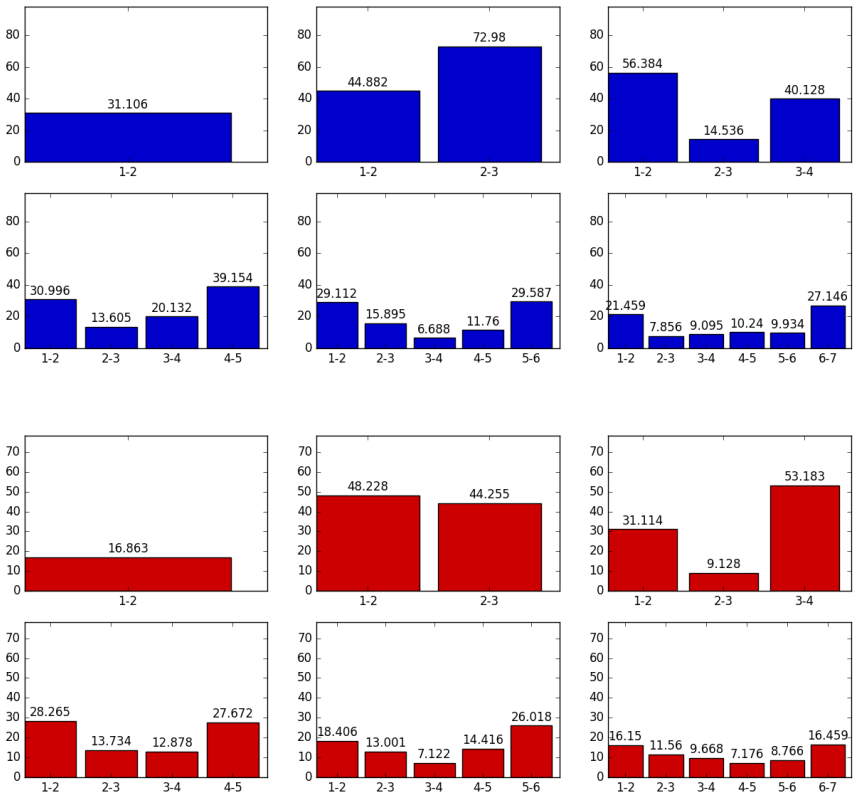
[3] Zdeněk Hanzlíček. *Czech HMM-Based Speech Synthesis*. *LNCS 6231*, pages 291 – 298, 2010.

[4] Alan W. Black Heiga Zen, Keiichi Tokuda. *Statistical parametric speech synthesis*. *Speech Communication 51*, pages 1039 – 1064, 2009.

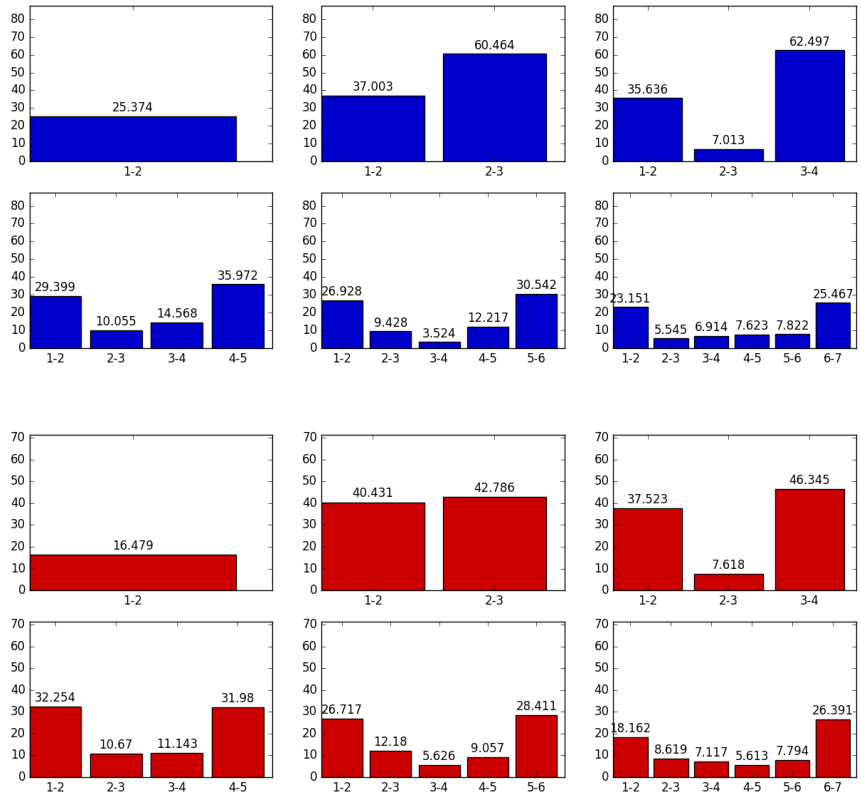
[5] John Duchi. *Derivations for Linear Algebra and Optimization*. Dostupné z: http://web.stanford.edu/~jduchi/projects/general_notes.pdf.

[6] M. J. F. Gales et al. S. J. Young, G. Evermann. *The HTK Book, version 3.4*, 2006.

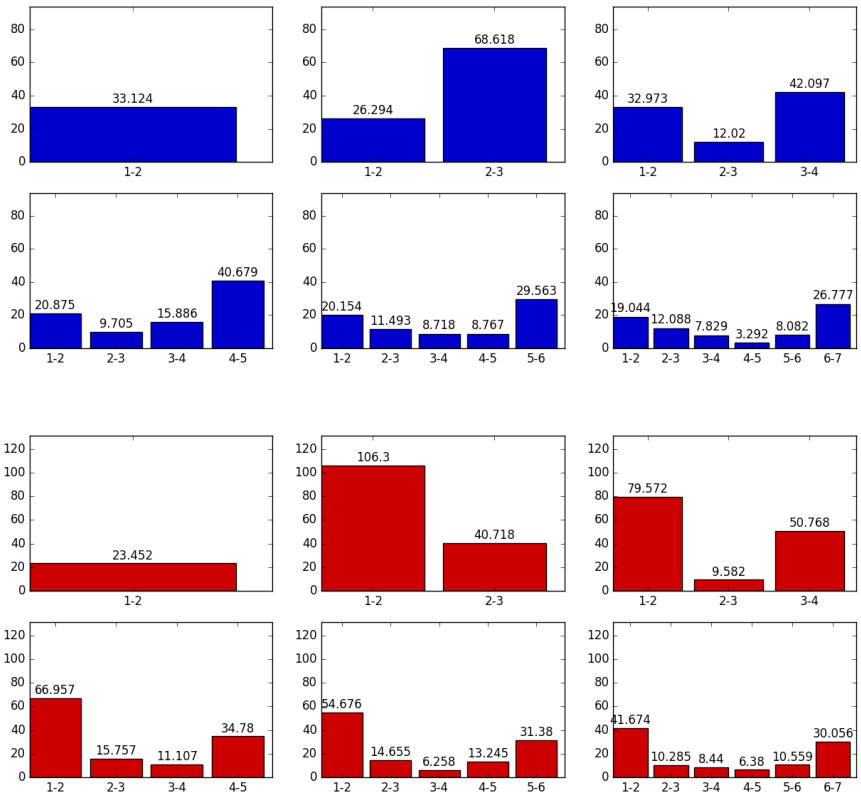
Přílohy



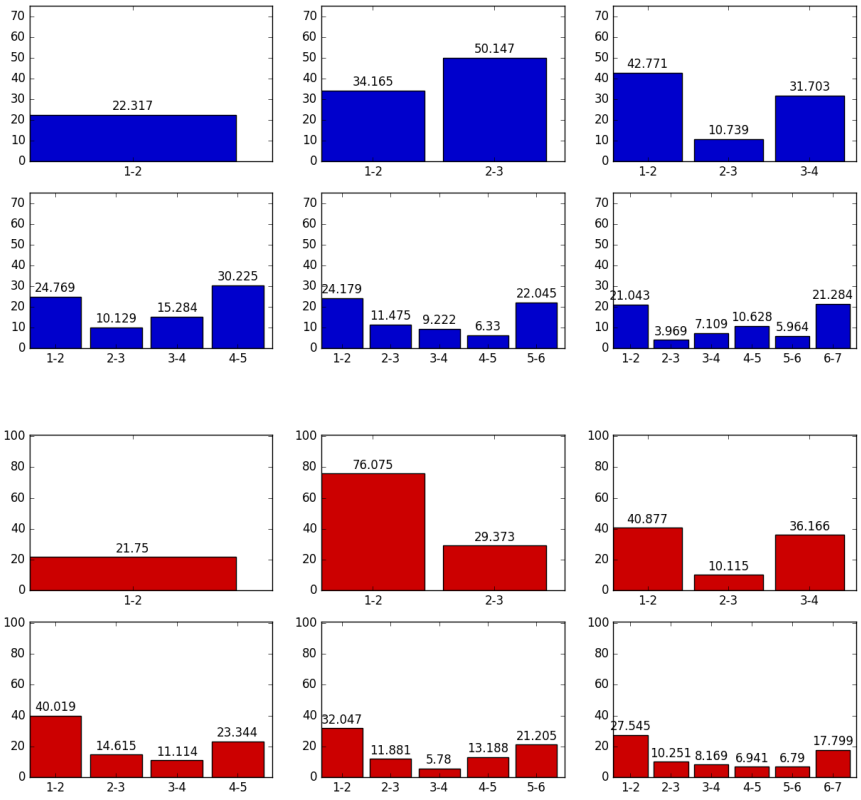
Hodnoty KL divergence pro monofon i



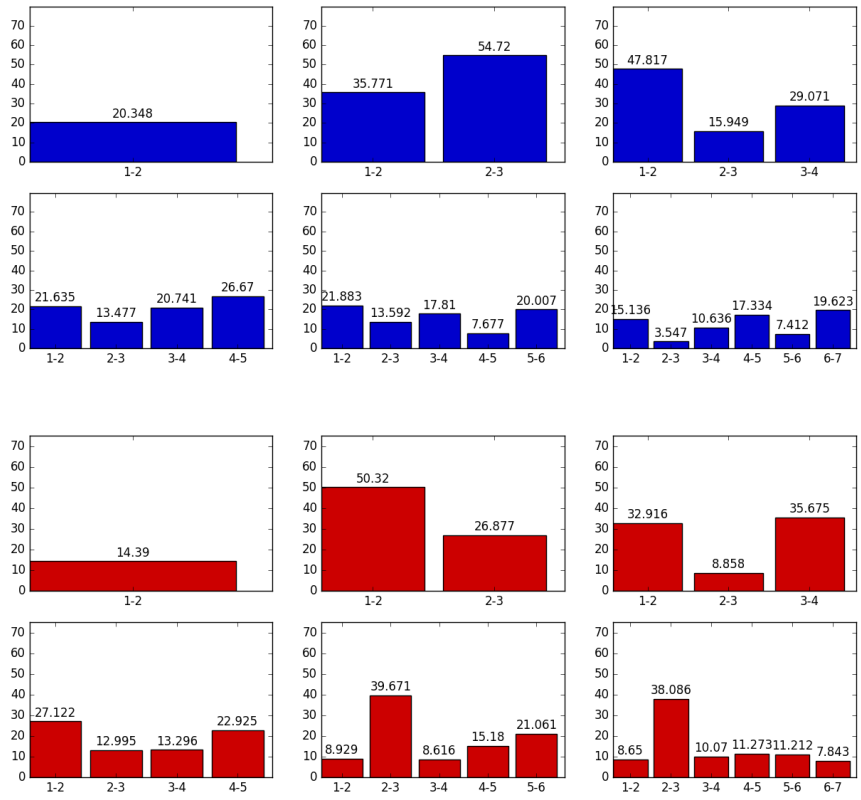
Hodnoty KL divergence pro monofon e



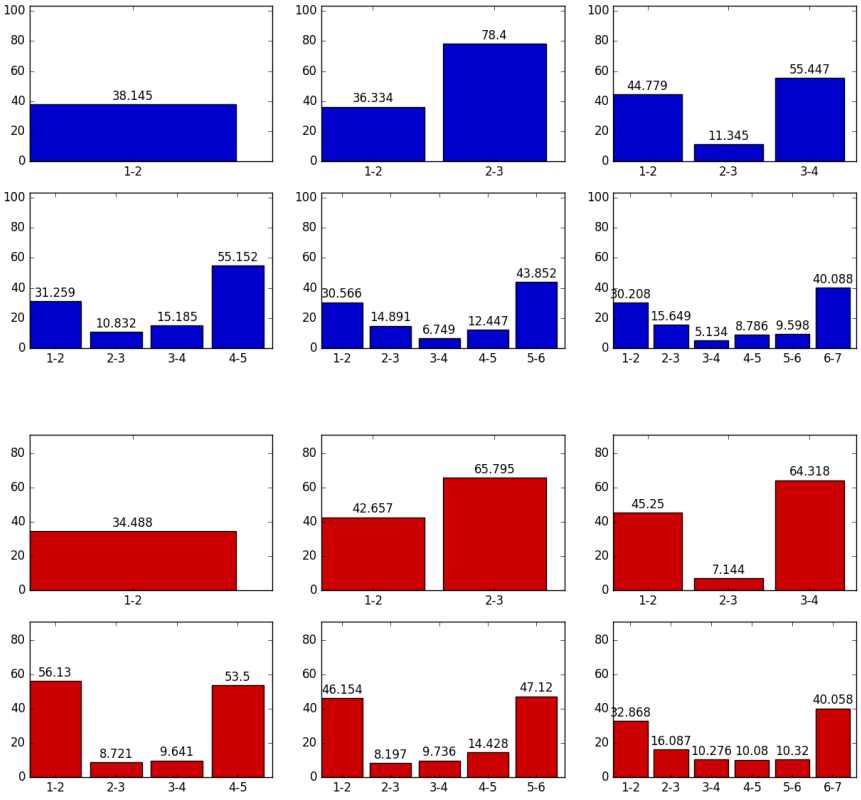
Hodnoty KL divergence pro monofon a



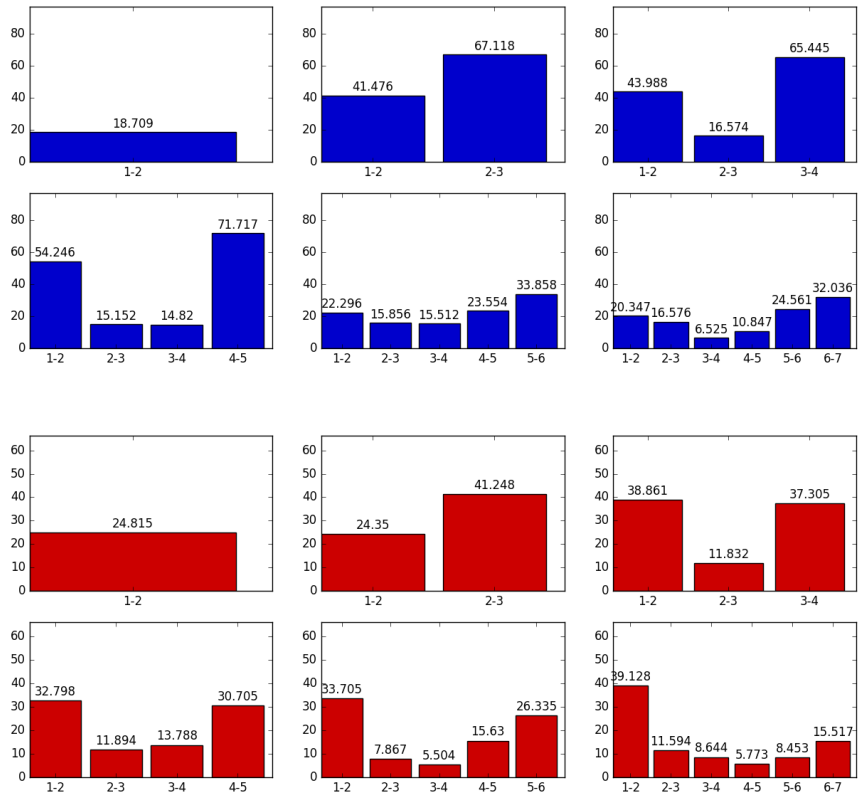
Hodnoty KL divergence pro monofon o



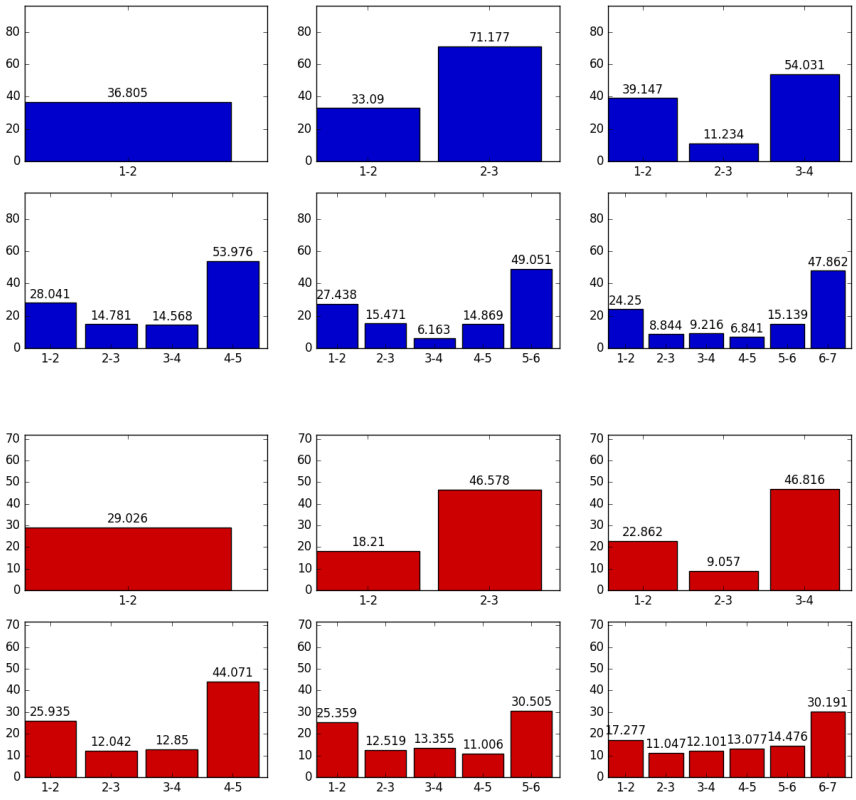
Hodnoty KL divergence pro monofon u



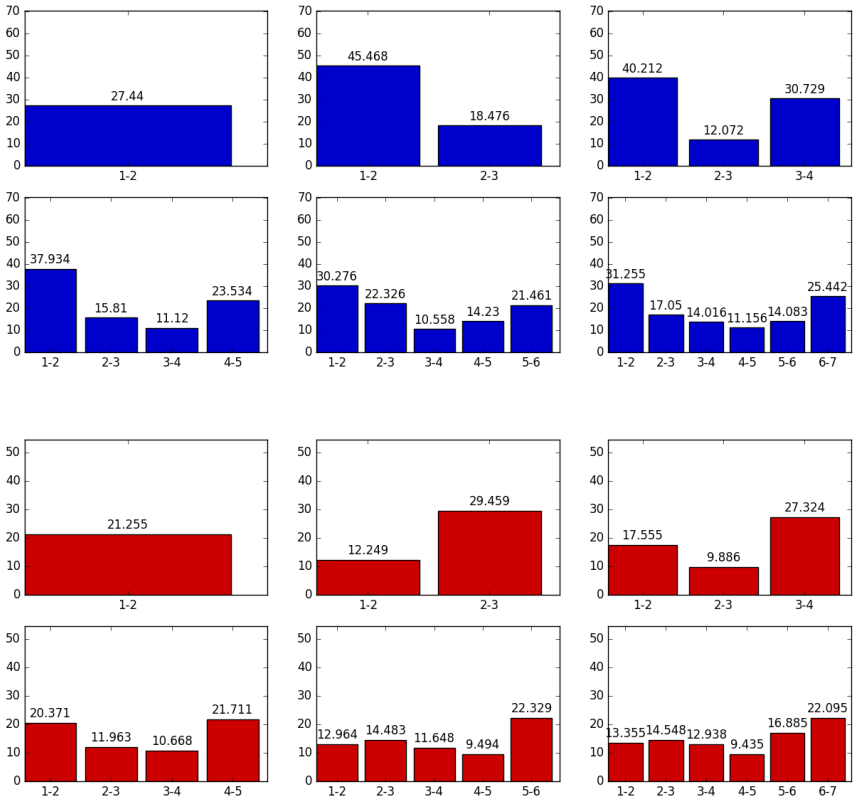
Hodnoty KL divergence pro monofon E



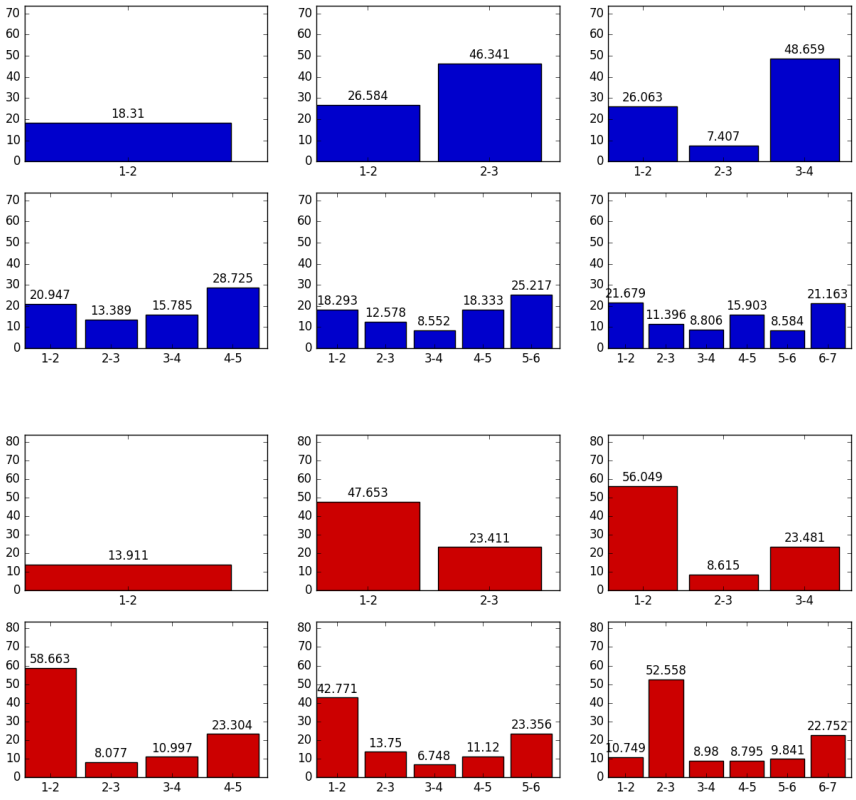
Hodnoty KL divergence pro monofon I



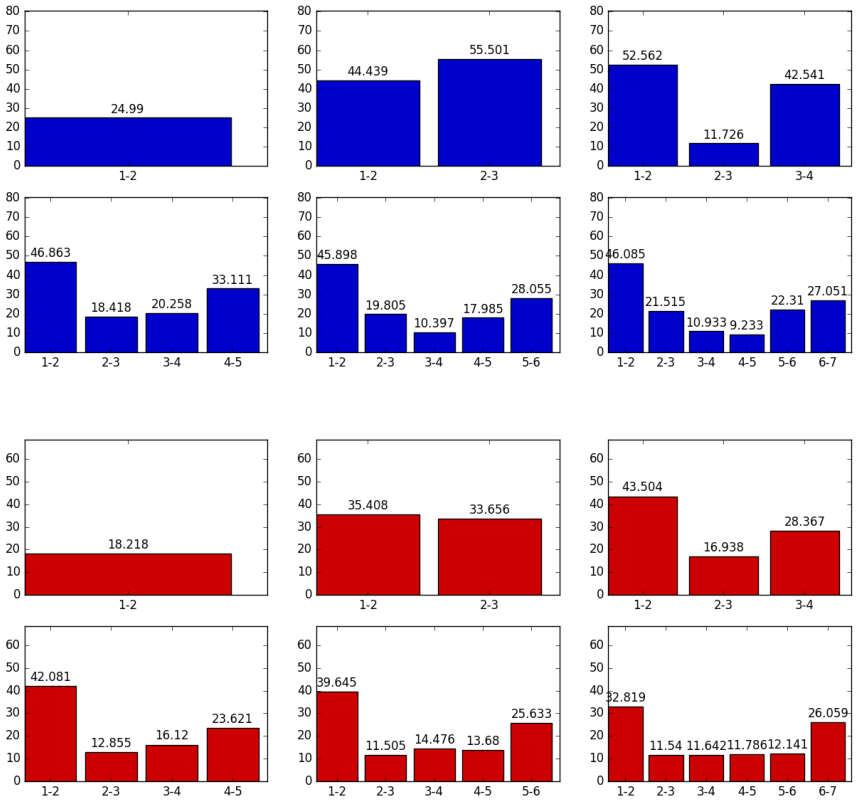
Hodnoty KL divergence pro monofon A



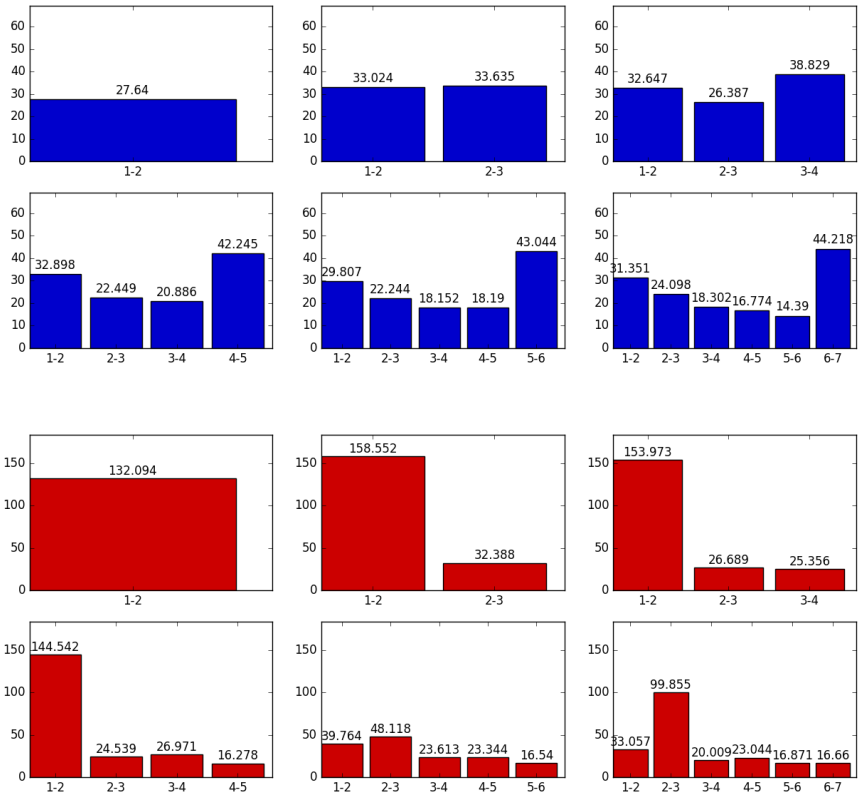
Hodnoty KL divergence pro monofon O



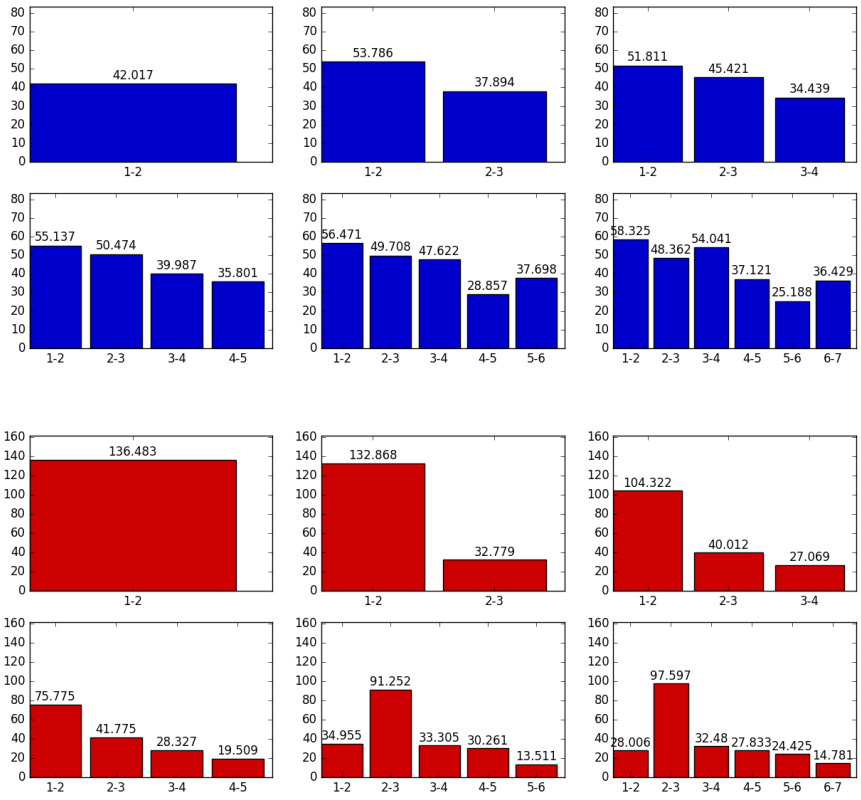
Hodnoty KL divergence pro monofon U



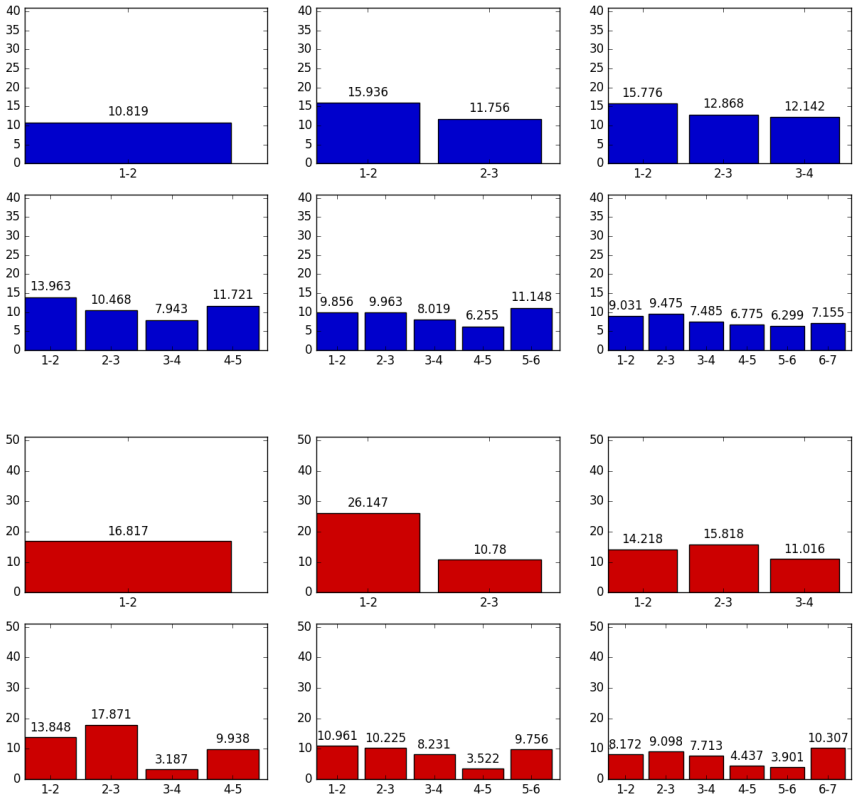
Hodnoty KL divergence pro monofon y



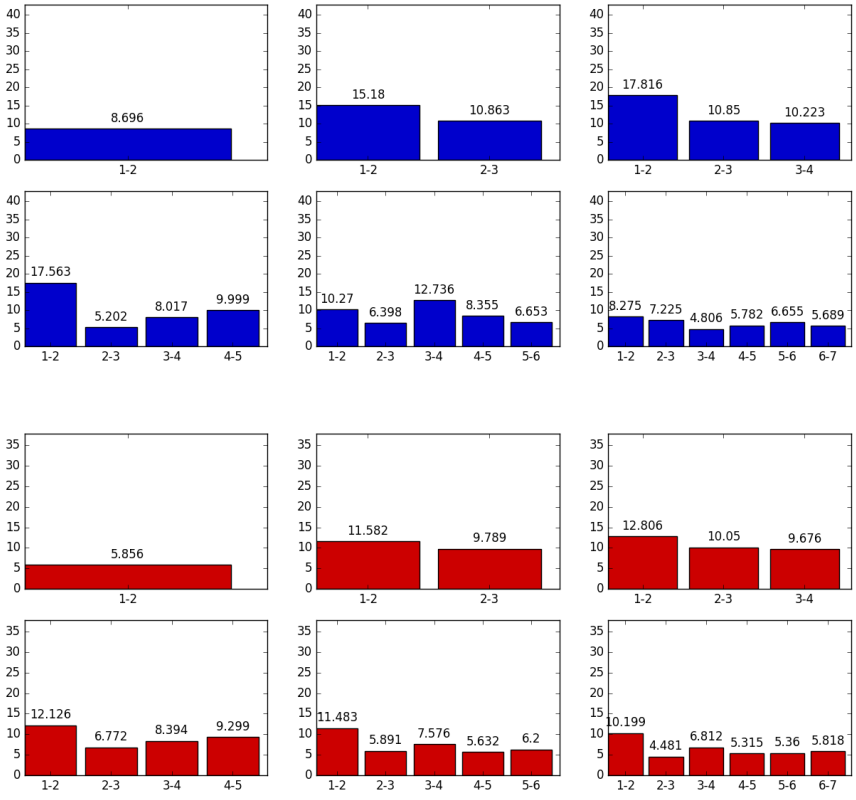
Hodnoty KL divergence pro monofon Y



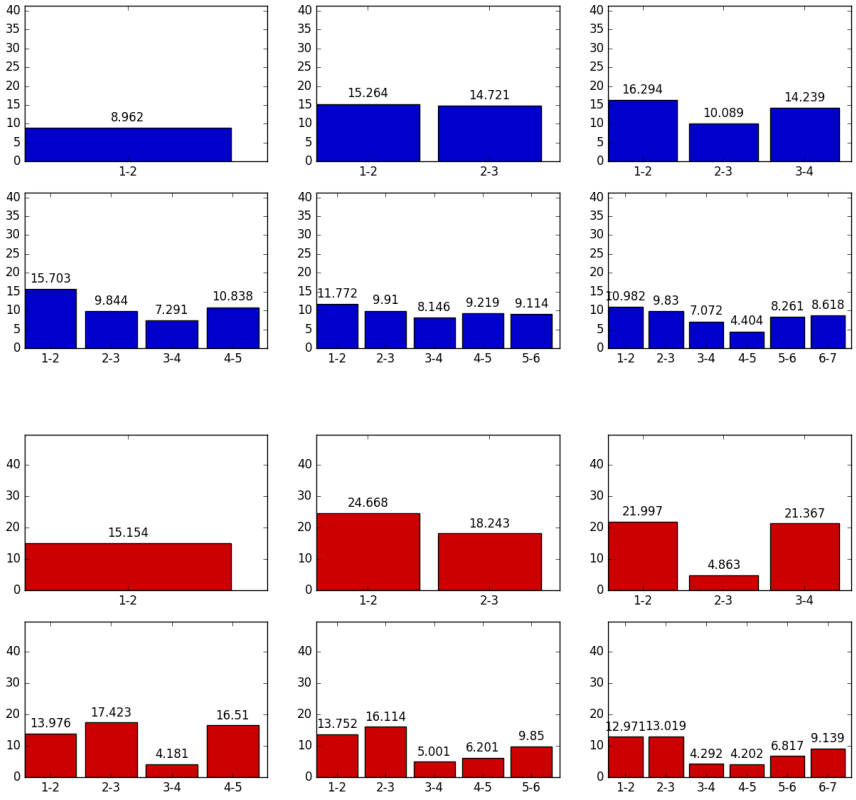
Hodnoty KL divergence pro monofon F



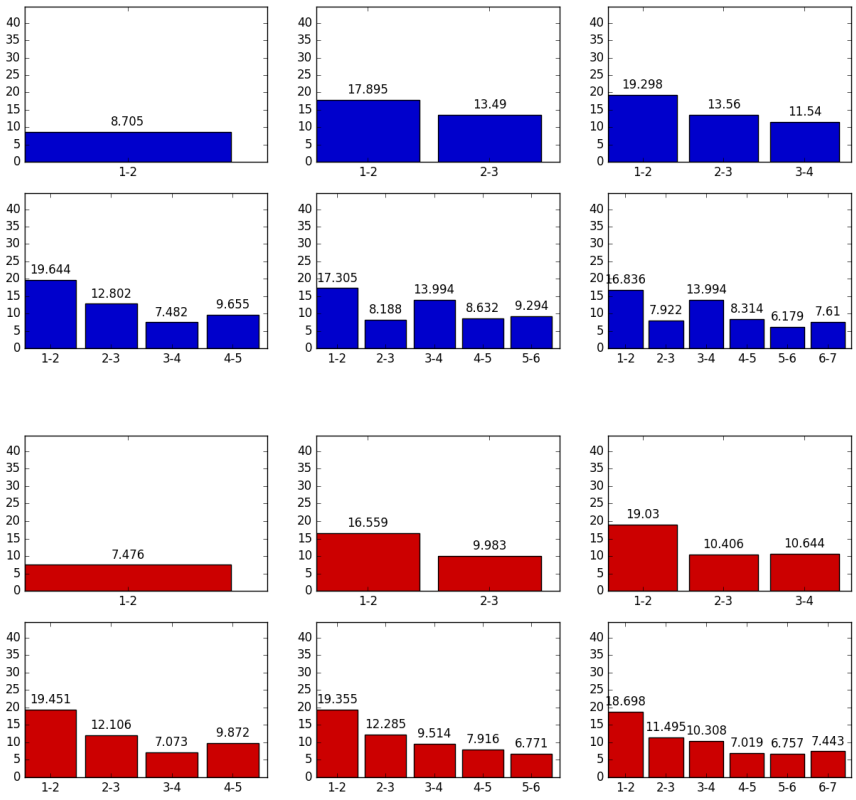
Hodnoty KL divergence pro monofon f



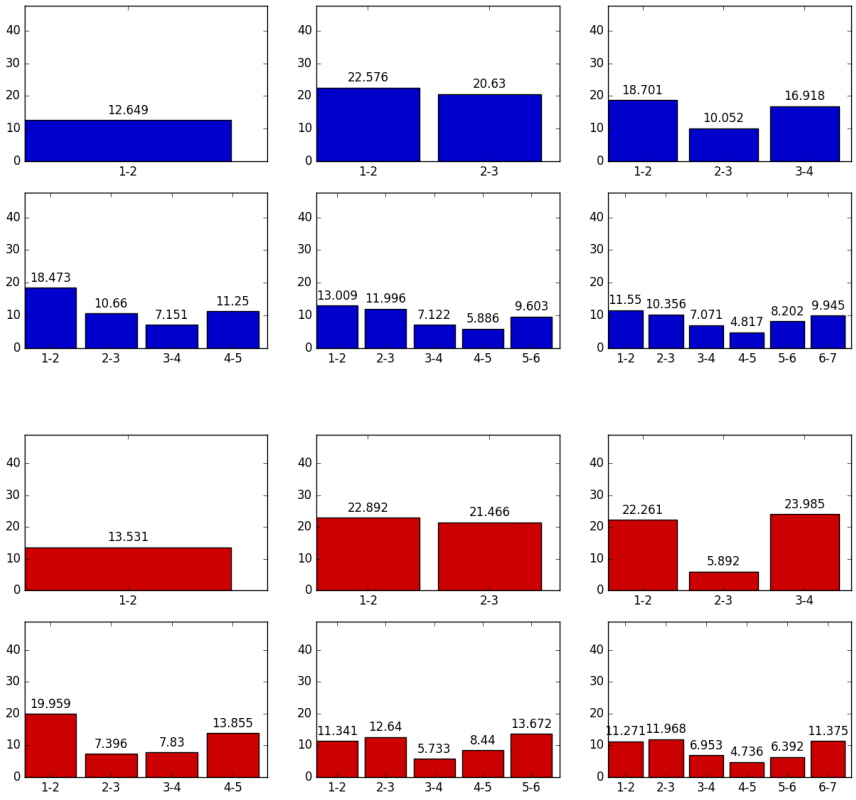
Hodnoty KL divergence pro monofon v



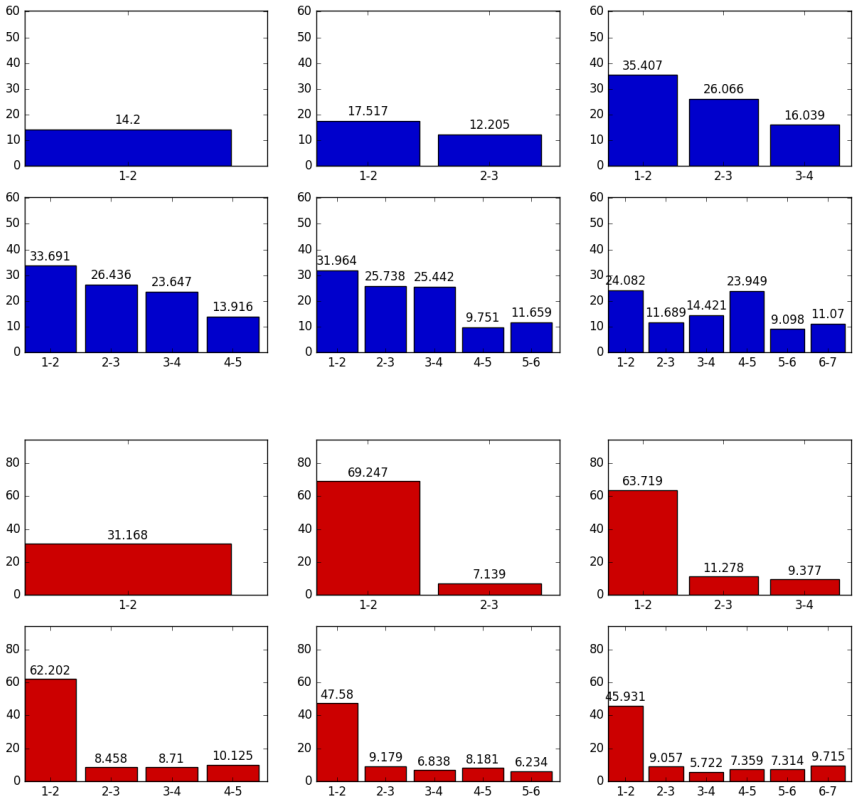
Hodnoty KL divergence pro monofon s



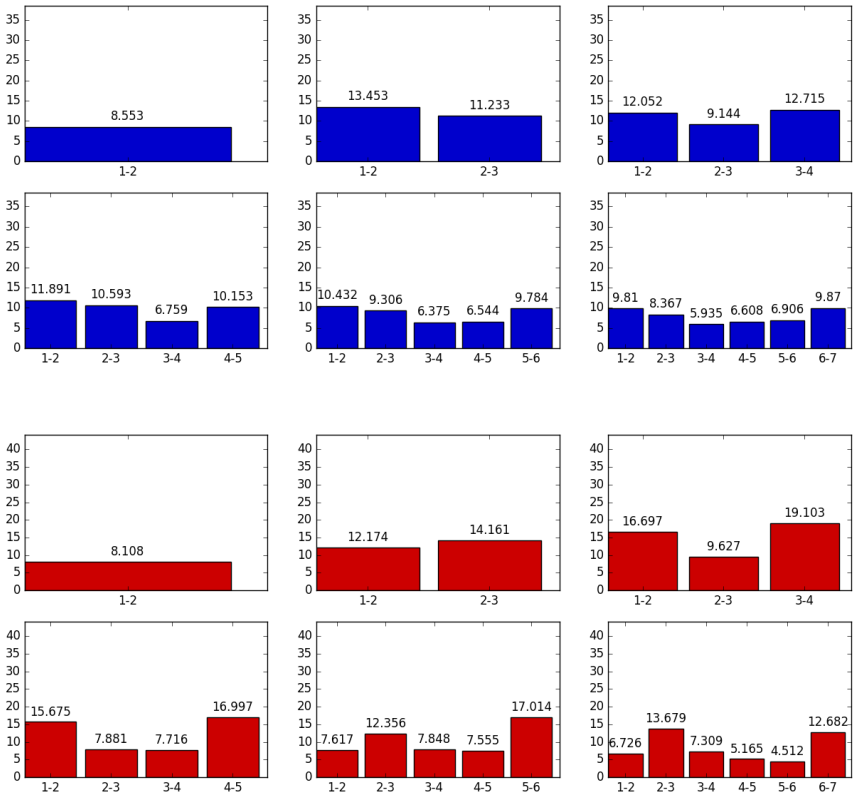
Hodnoty KL divergence pro monofon z



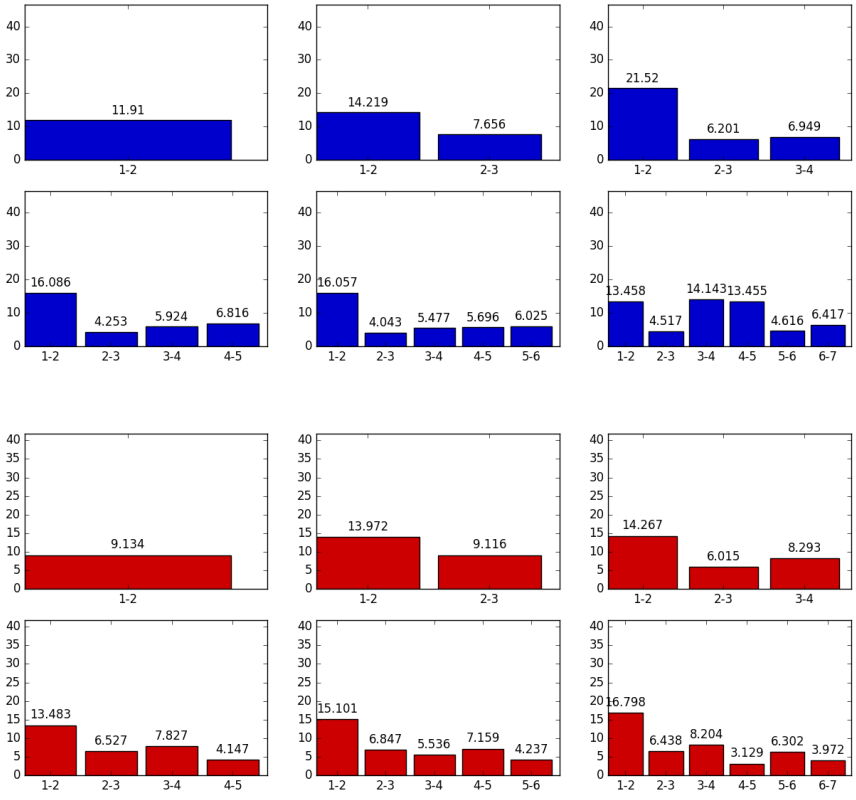
Hodnoty KL divergence pro monofon S



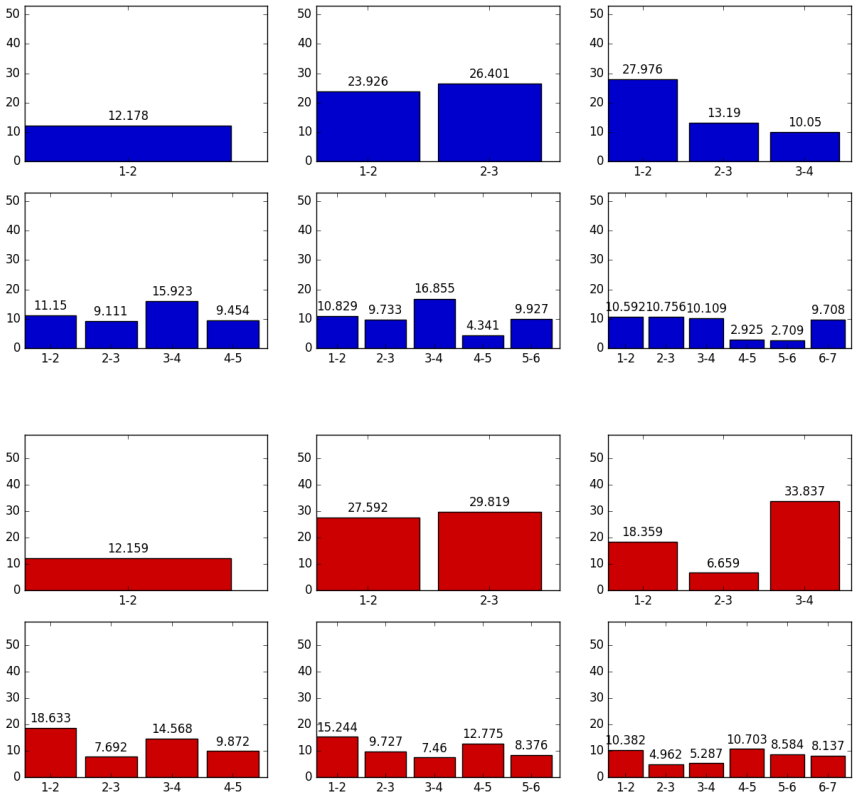
Hodnoty KL divergence pro monofon Z



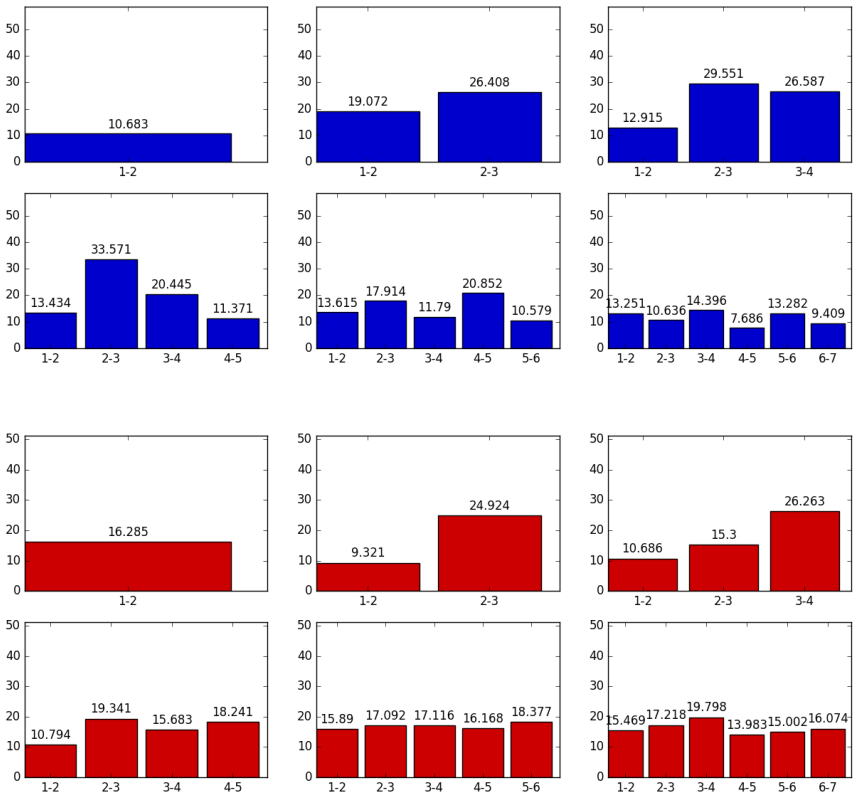
Hodnoty KL divergence pro monofon x



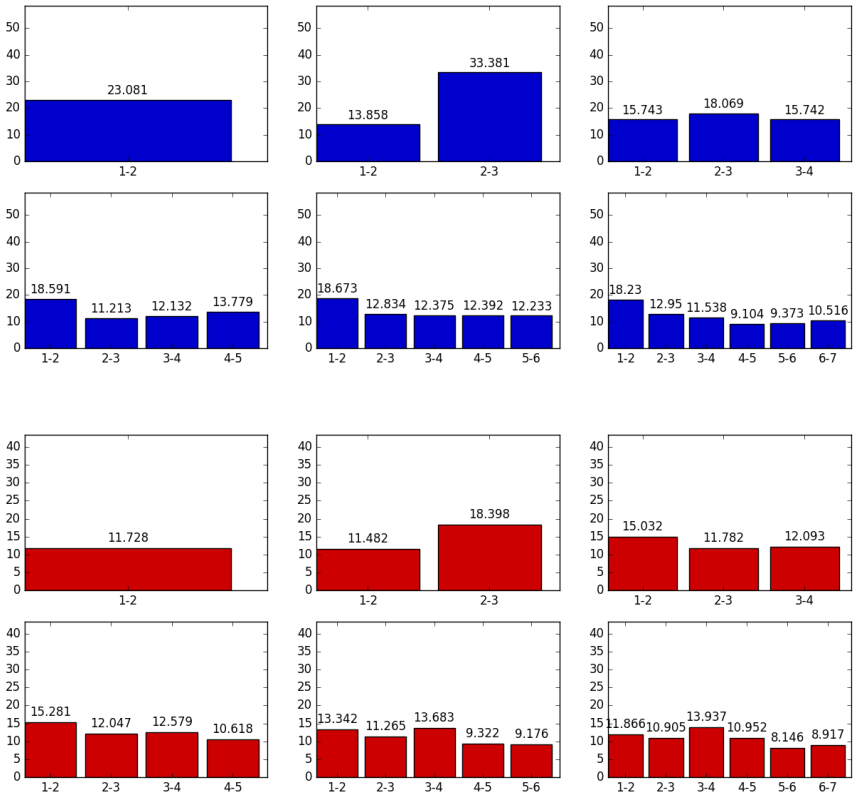
Hodnoty KL divergence pro monofon h



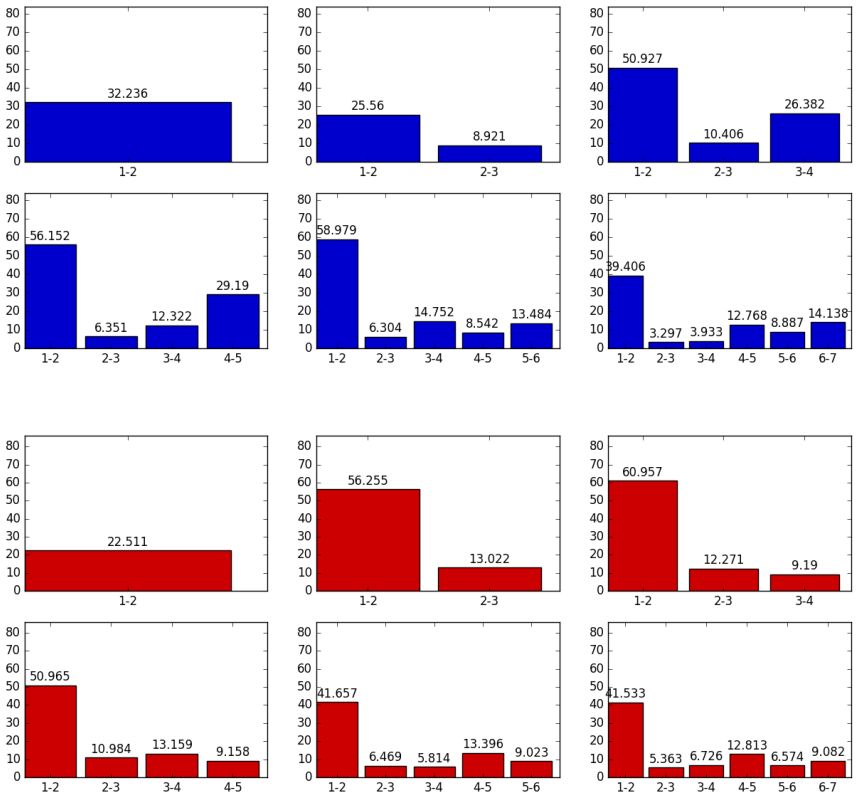
Hodnoty KL divergence pro monofon l



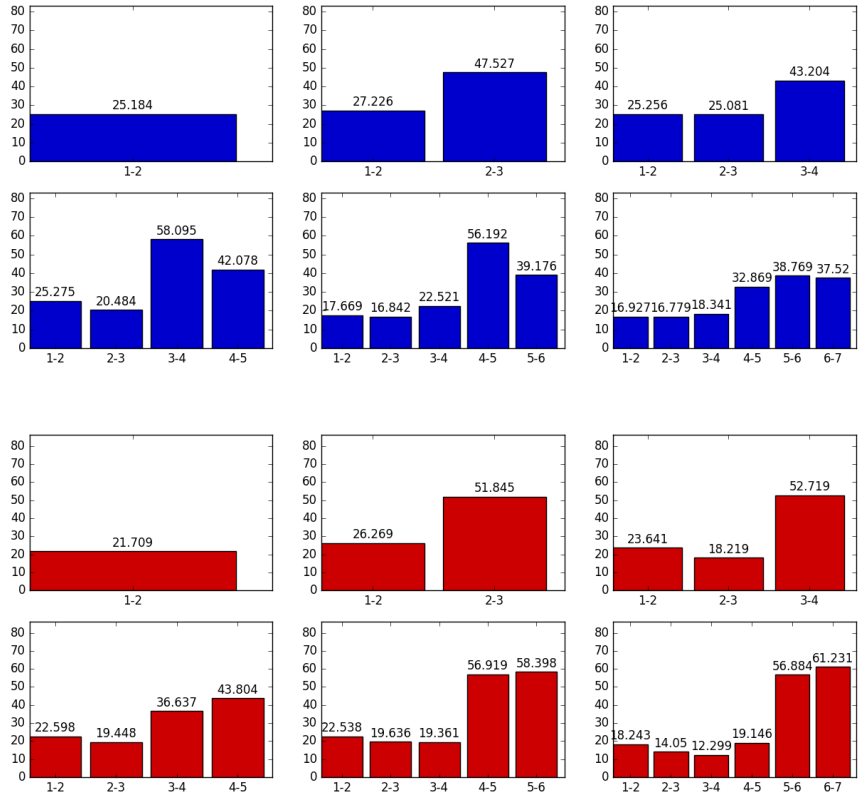
Hodnoty KL divergence pro monofon r



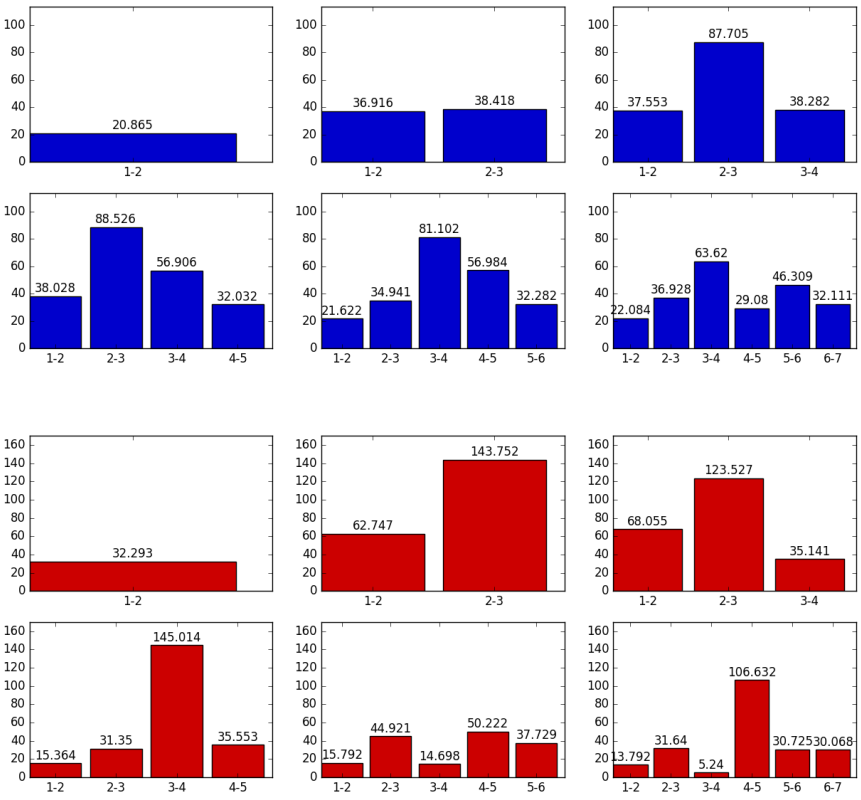
Hodnoty KL divergence pro monofon R



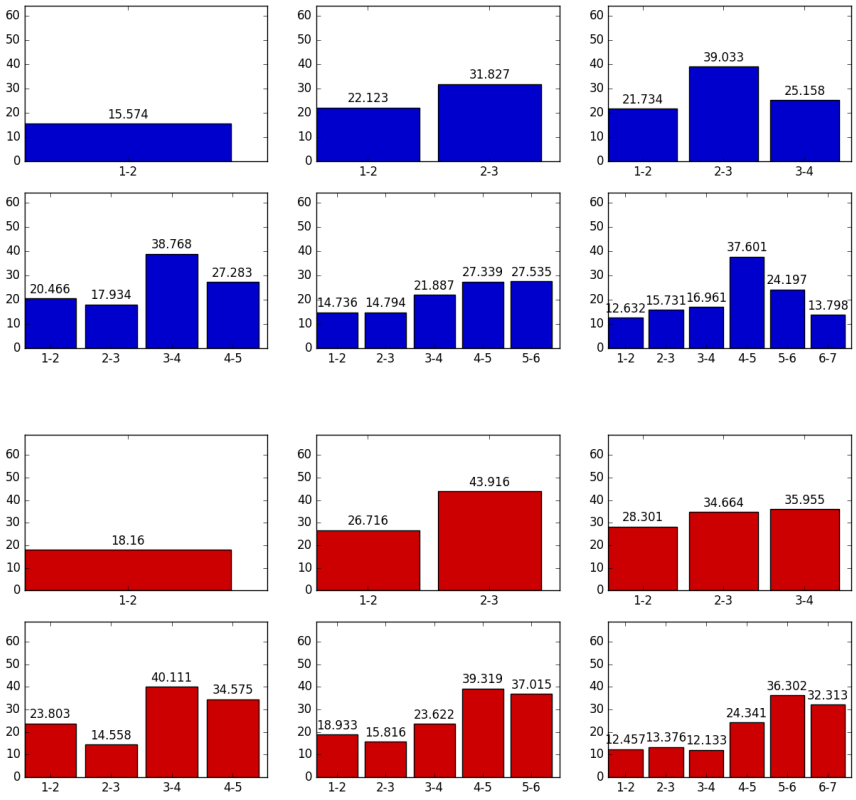
Hodnoty KL divergence pro monofon j



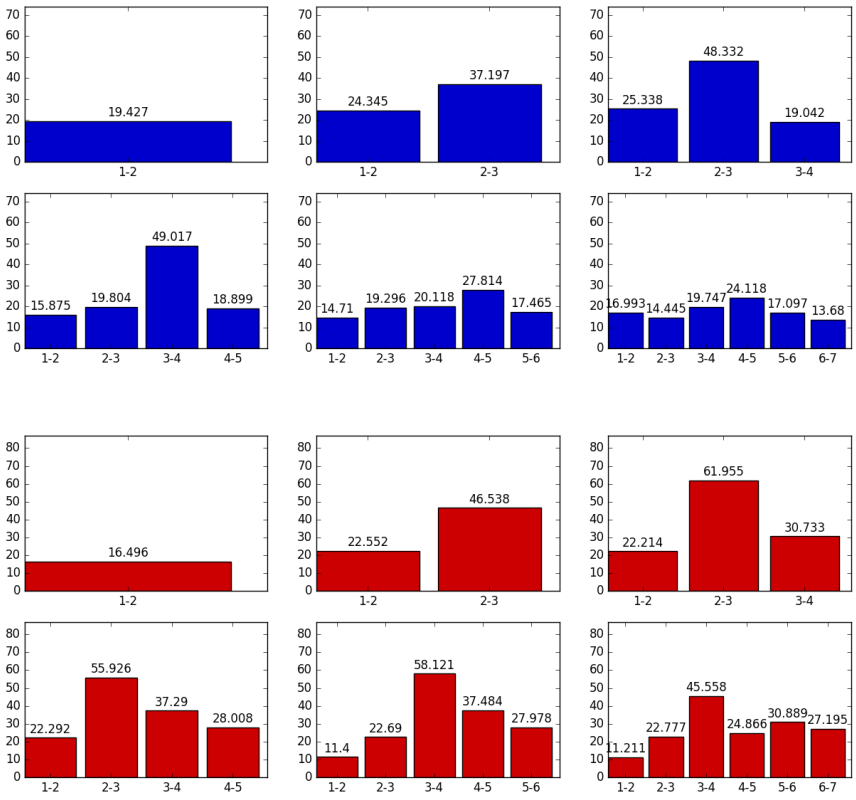
Hodnoty KL divergence pro monofon p



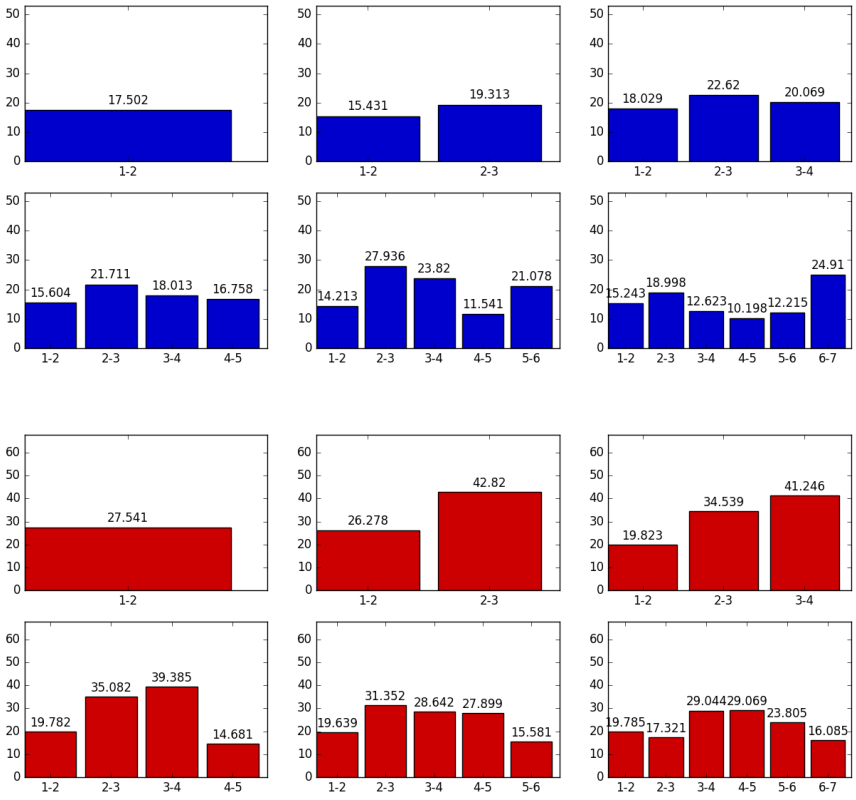
Hodnoty KL divergence pro monofon b



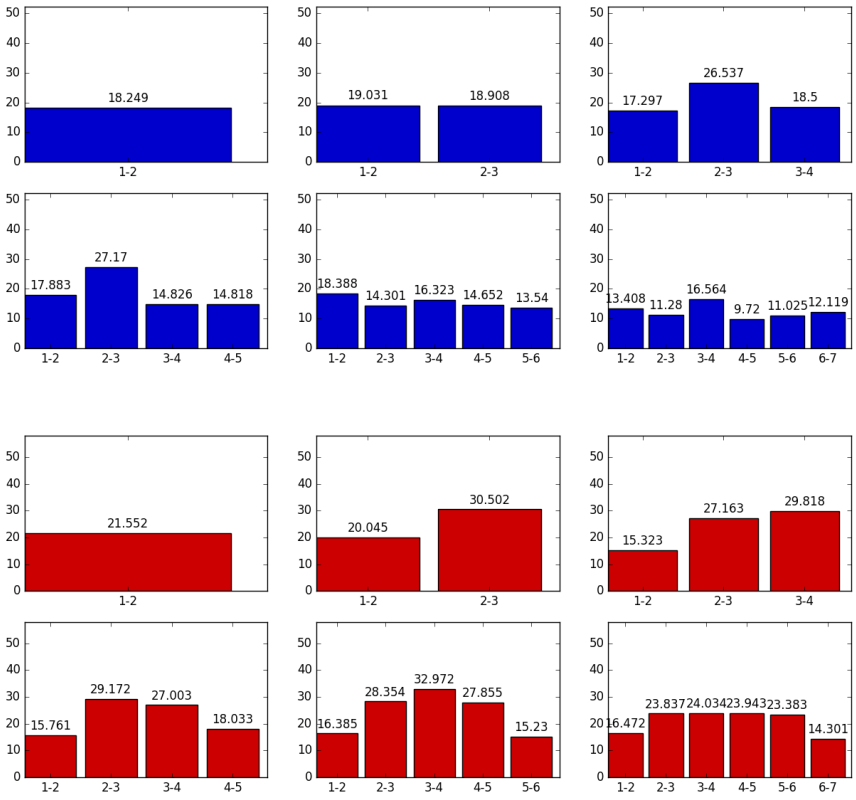
Hodnoty KL divergence pro monofon t



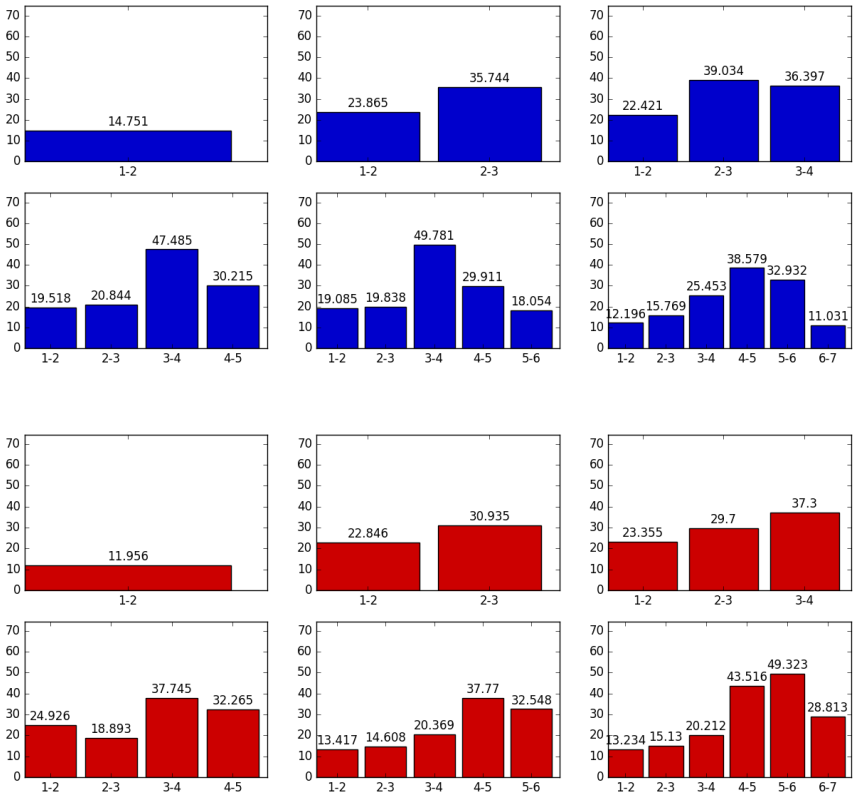
Hodnoty KL divergence pro monofon d



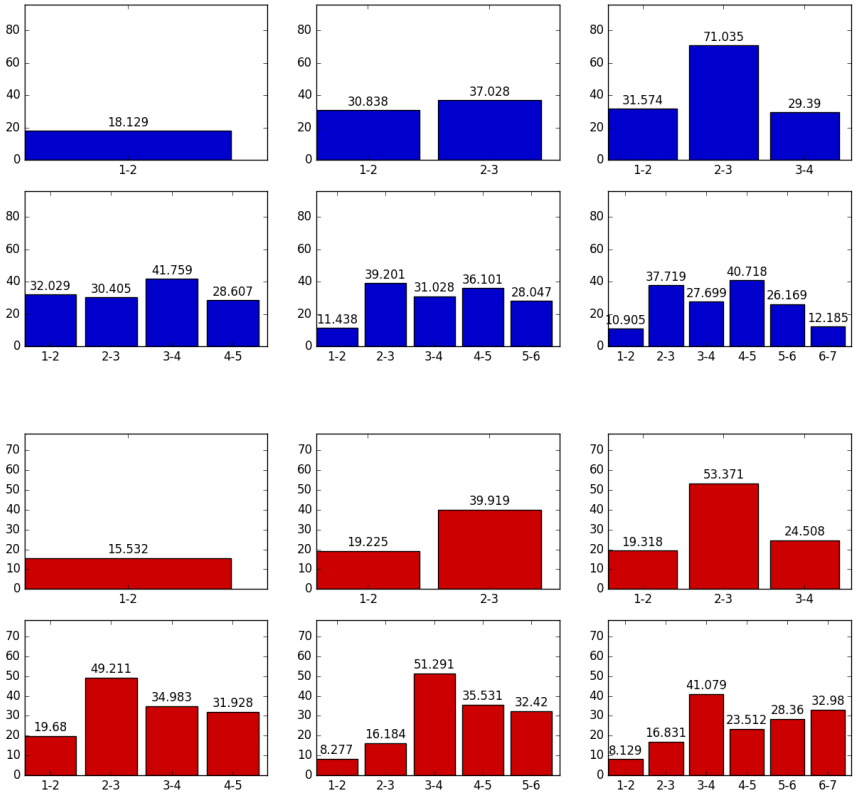
Hodnoty KL divergence pro monofon T



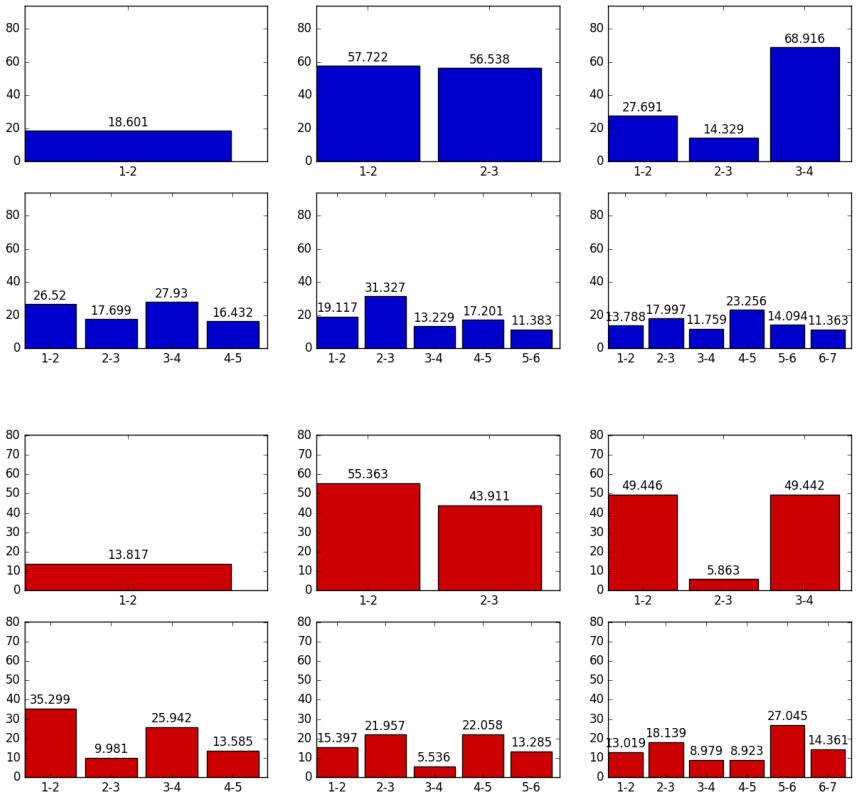
Hodnoty KL divergence pro monofon d



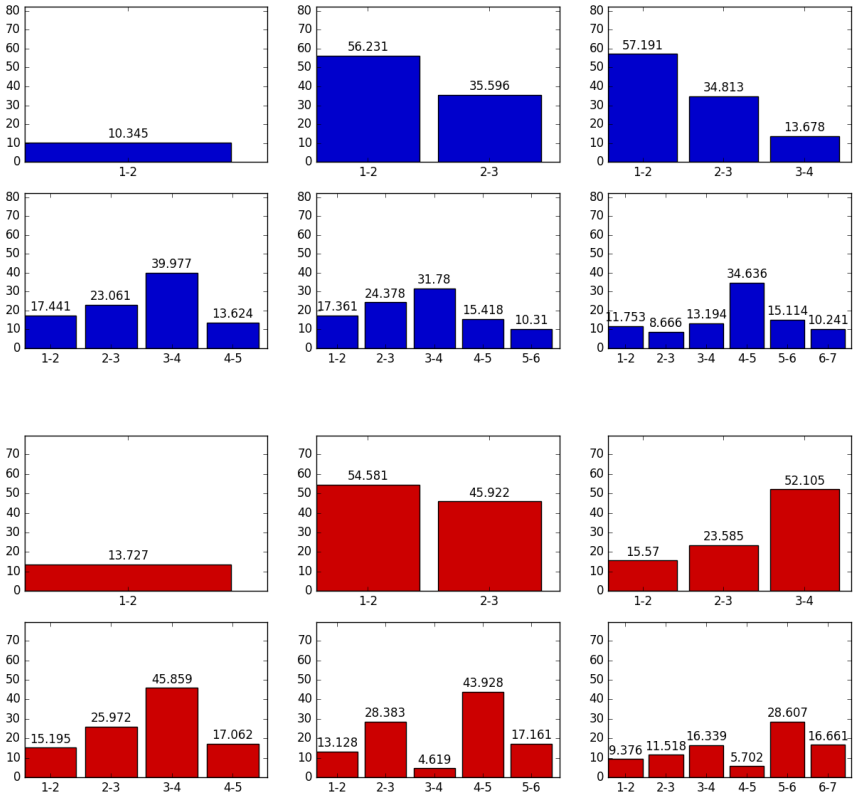
Hodnoty KL divergence pro monofon k



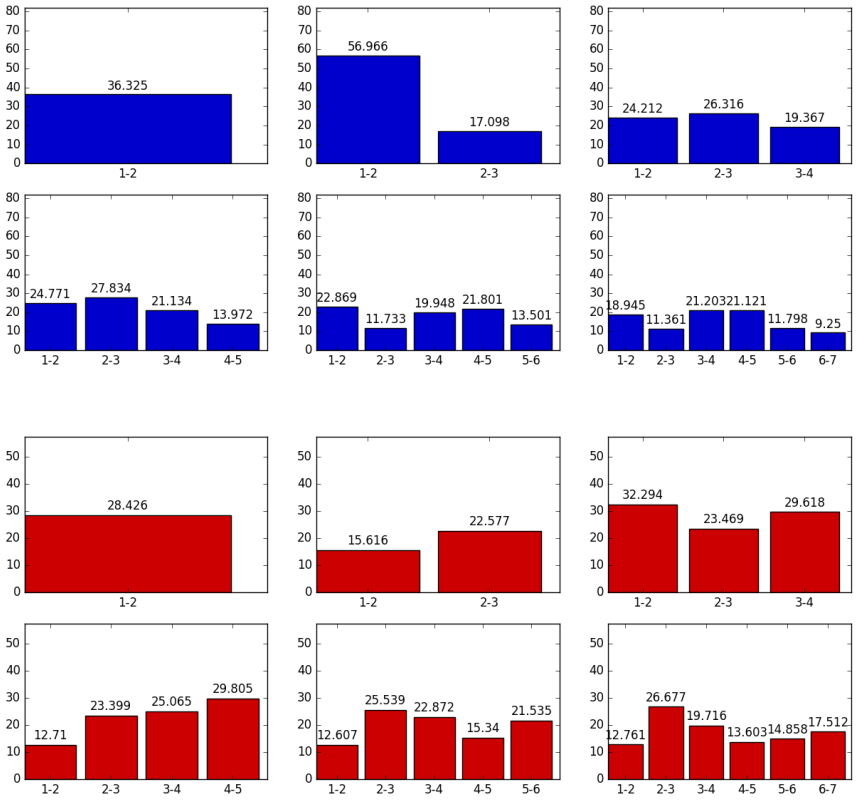
Hodnoty KL divergence pro monofon g



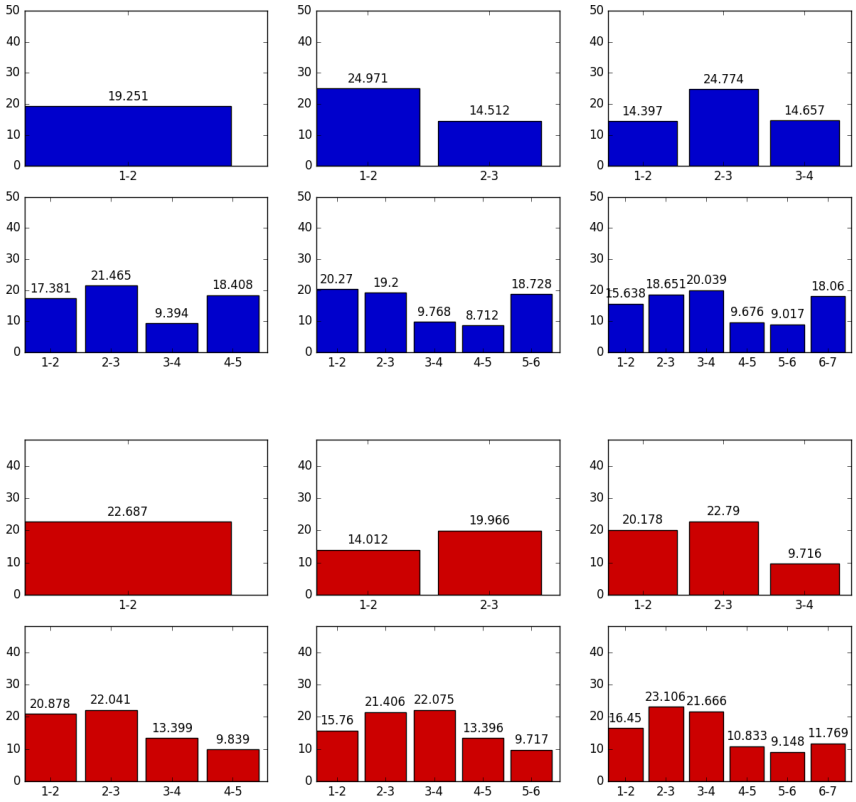
Hodnoty KL divergence pro monofon m



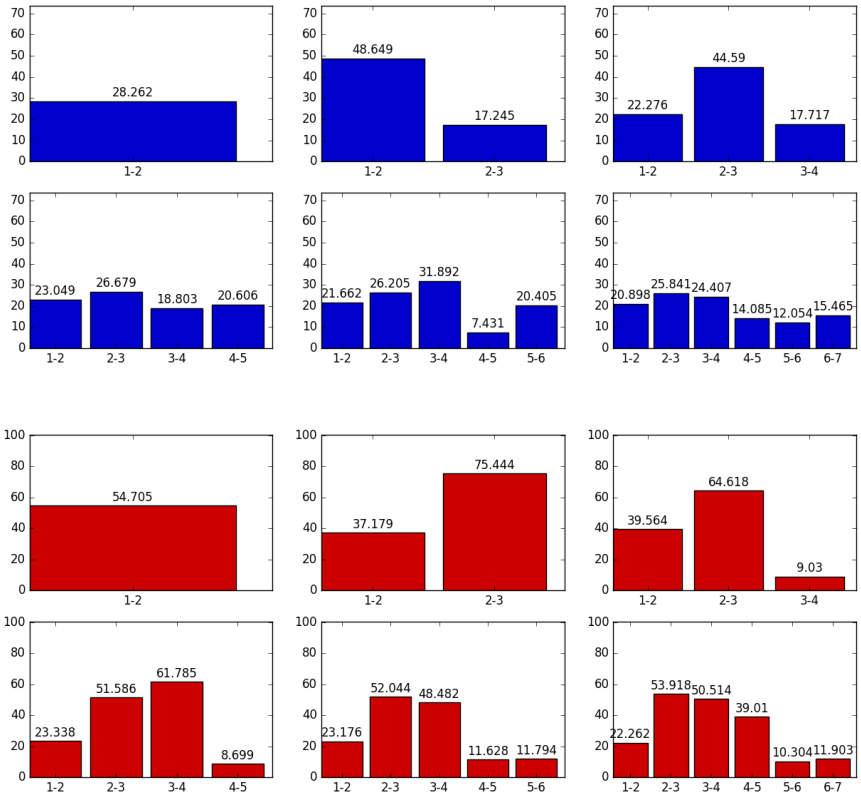
Hodnoty KL divergence pro monofon n



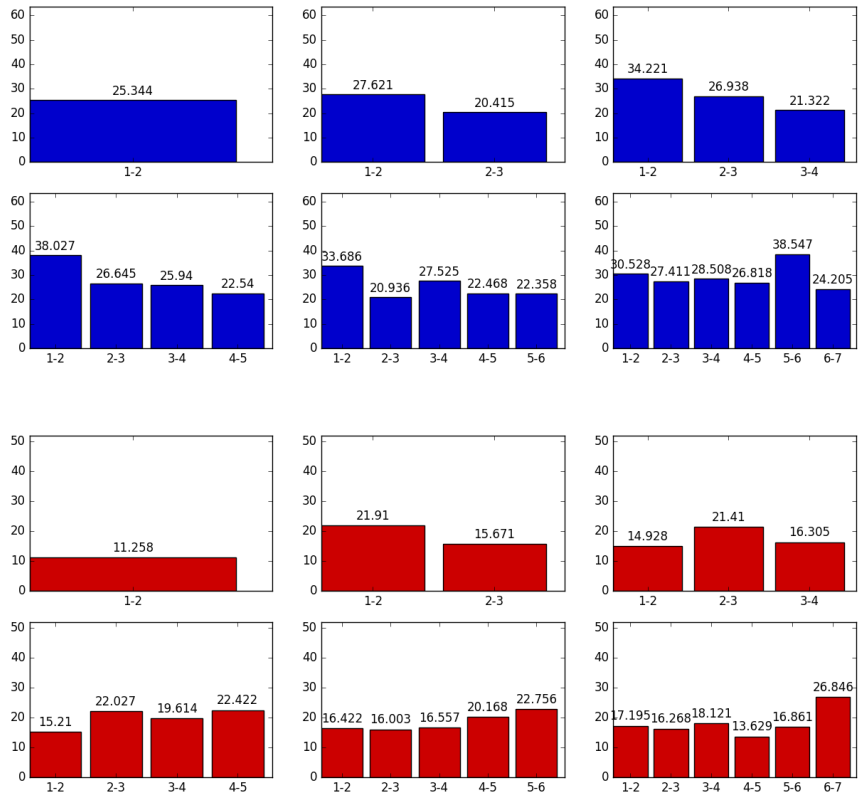
Hodnoty KL divergence pro monofon J



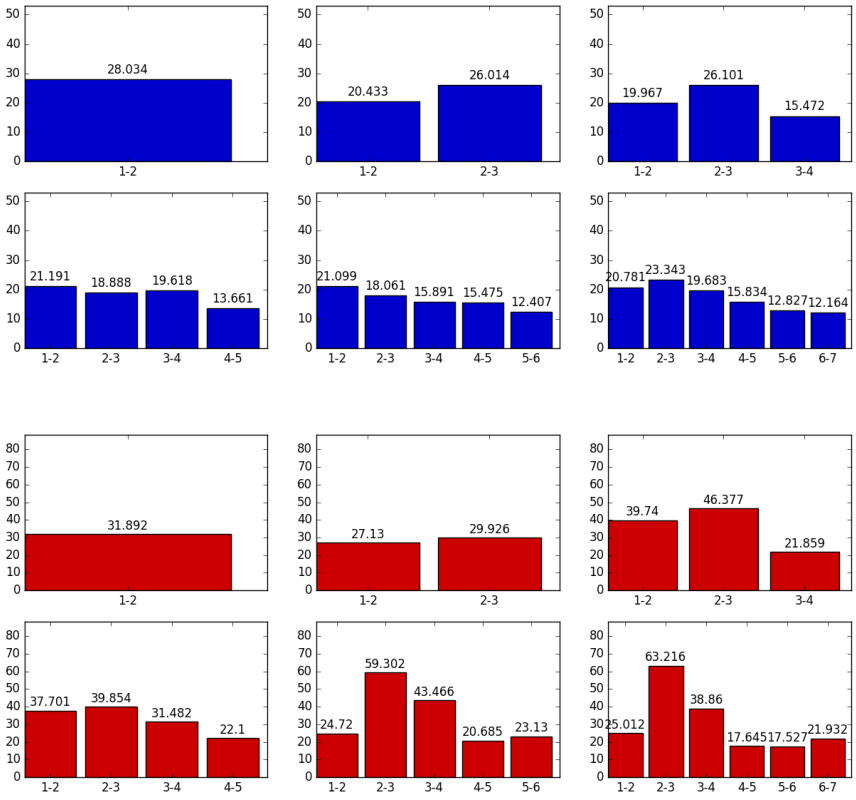
Hodnoty KL divergence pro monofon c



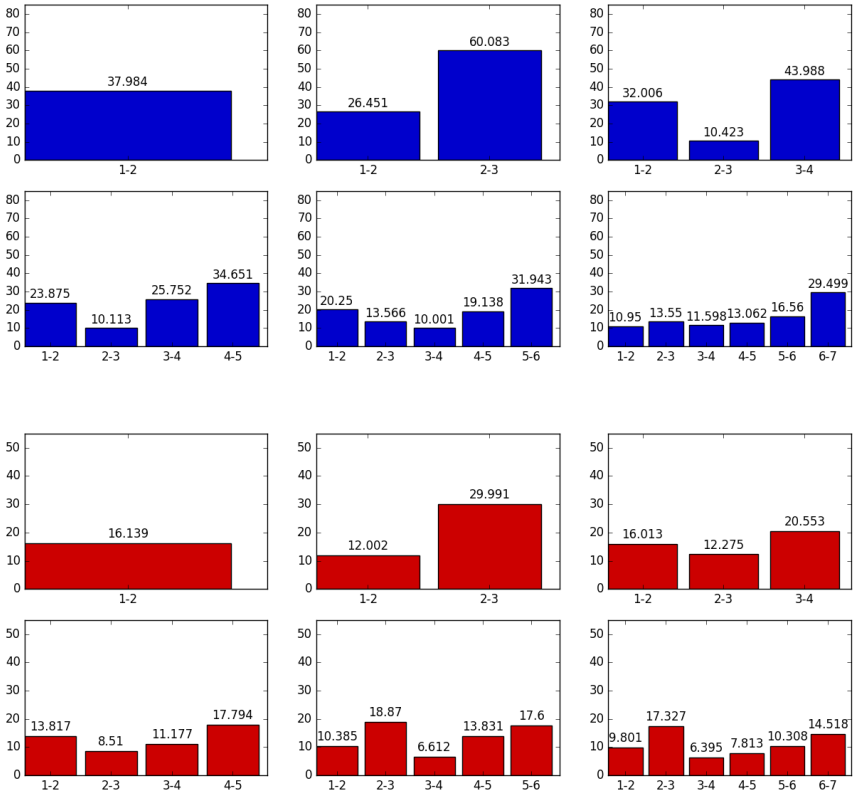
Hodnoty KL divergence pro monofon C



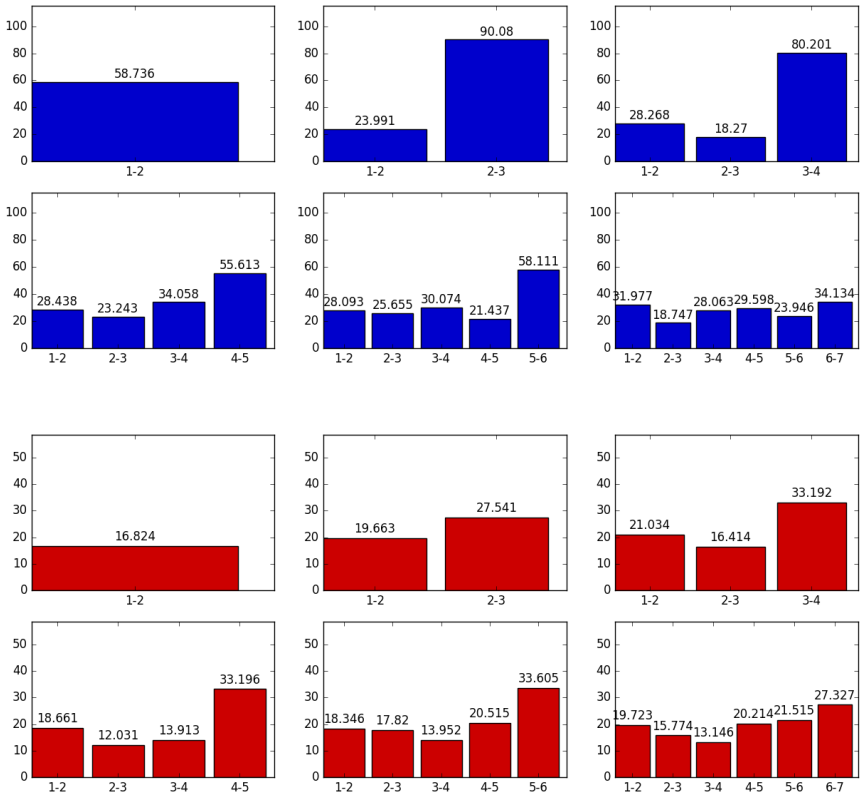
Hodnoty KL divergence pro monofon w



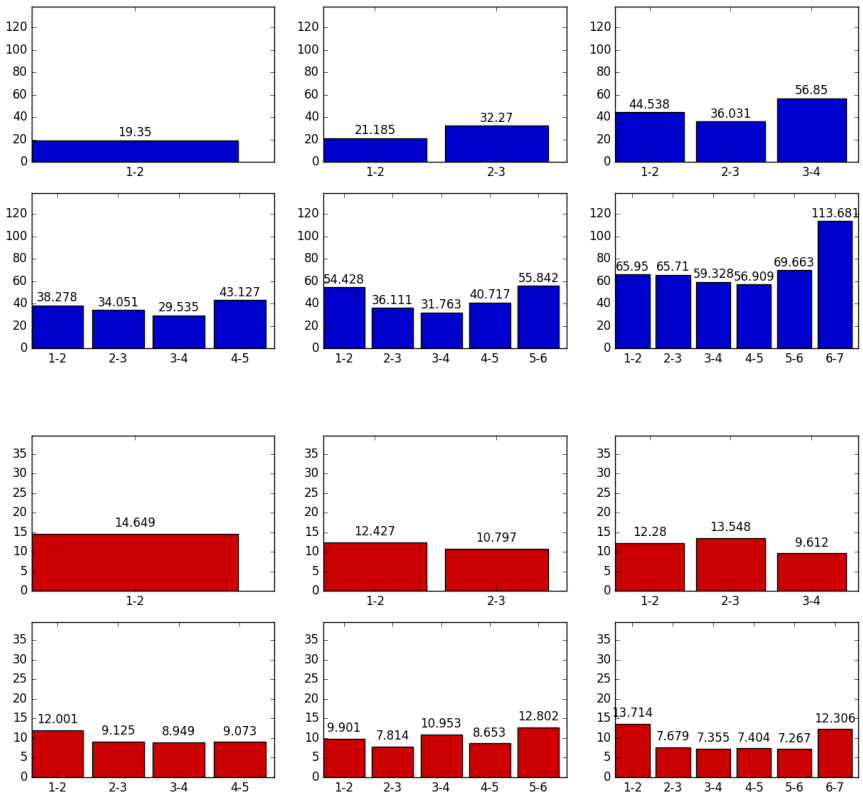
Hodnoty KL divergence pro monofon W



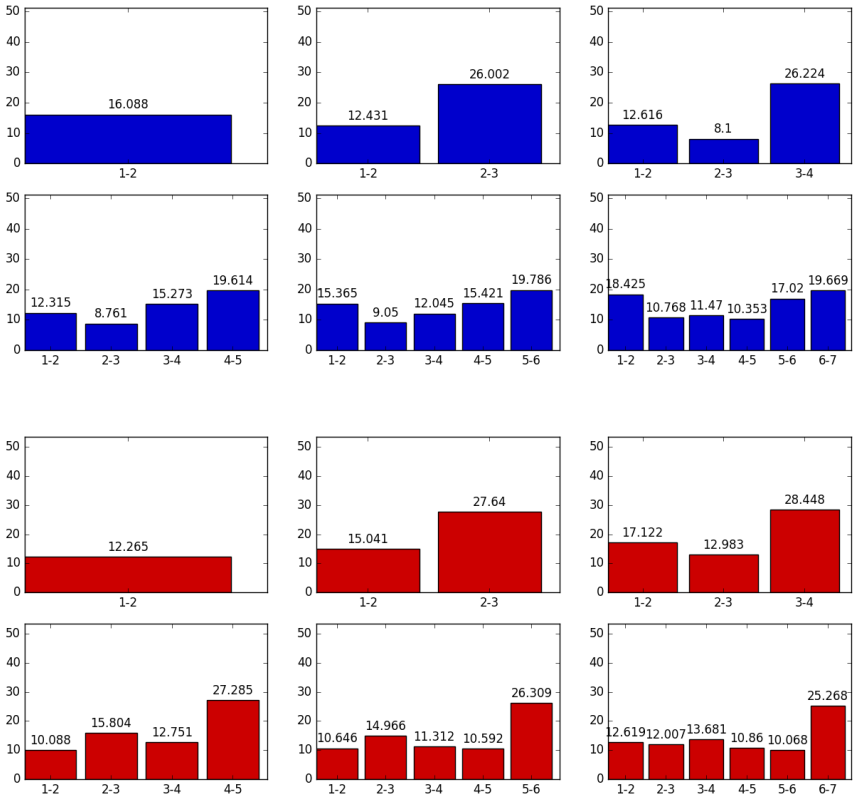
Hodnoty KL divergence pro monofon N



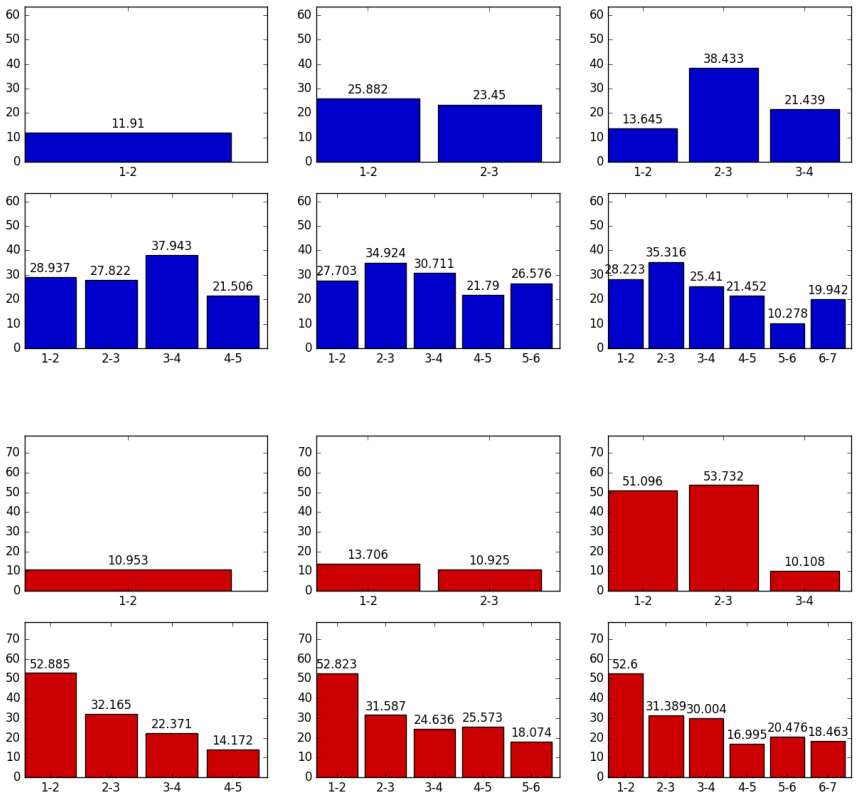
Hodnoty KL divergence pro monofon M



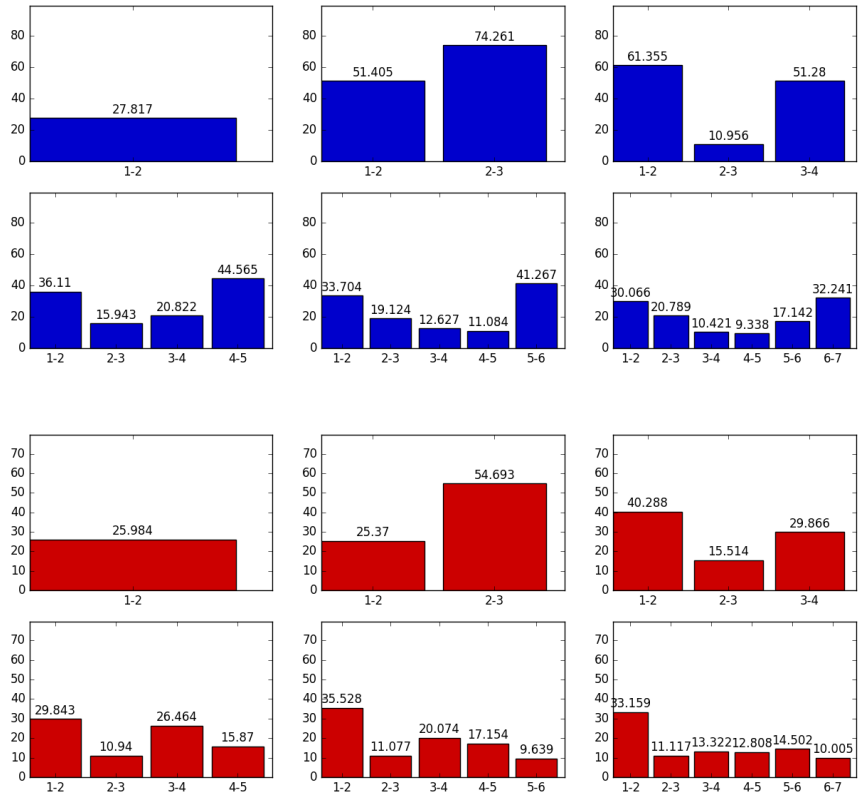
Hodnoty KL divergence pro monofon G



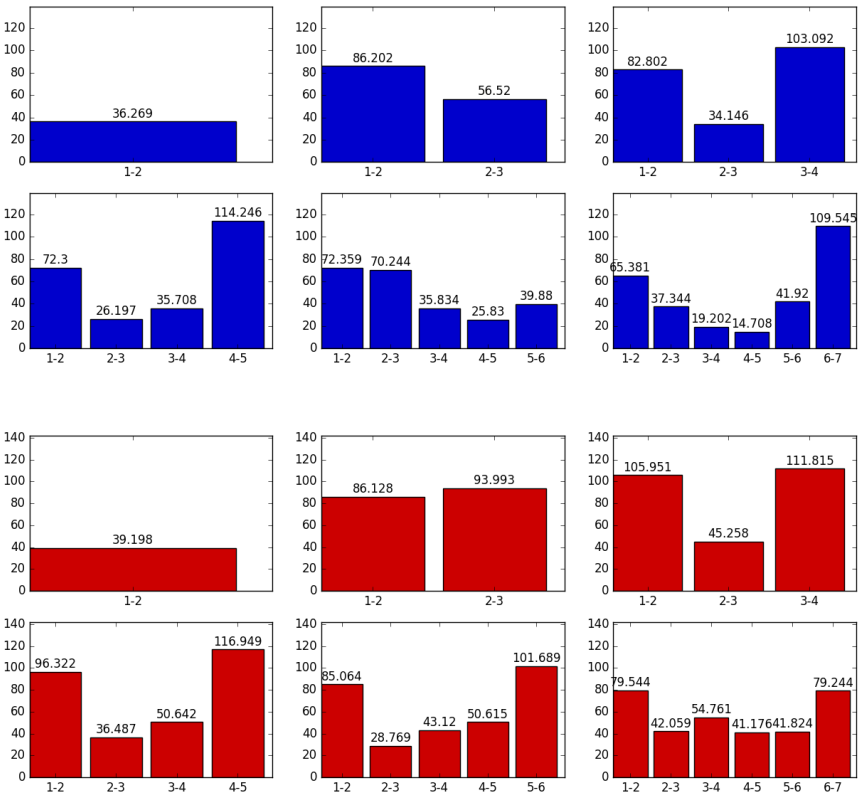
Hodnoty KL divergence pro monofon Q



Hodnoty KL divergence pro monofon P



Hodnoty KL divergence pro monofon L



Hodnoty KL divergence pro monofon H