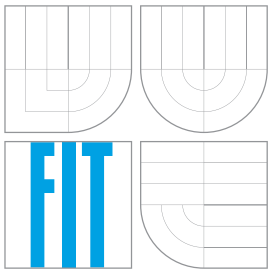
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ADAPTACE SYSTÉMŮ PRO ROZPOZNÁNÍ MLUVČÍHO

ADAPTATION OF SPEAKER RECOGNITION SYSTEMS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ONDŘEJ NOVOTNÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. OLDŘICH PLCHOT

BRNO 2014

Abstrakt

V této práci navrhneme techniky adaptace systémů na rozpoznávání řeči. Cílem je vytvořit techniku adaptace Pravděpodobnostní lineární diskriminační analýzy. Zaměříme se na adaptaci bez učitele. Naše testy ukáží vhodné shlukovací techniky pro odhad identity mluvčích a vhodné techniky na odhad počtu mluvčích v adaptační datové sadě. Experimenty jsou prováděny na korpusech NIST a Switchboard.

Abstract

In this paper, we propose techniques for adaptation of speaker recognition systems. The aim of this work is to create adaptation for Probabilistic Linear Discriminant Analysis. Special attention is given to unsupervised adaptation. Our test shows appropriate clustering techniques for speaker estimation of the identity and estimation of the number of speakers in adaptation dataset. For the test, we are using NIST and Switchboard corpora.

Klíčová slova

Rozpoznávání, rozpoznávání mluvčích, shlukování, adaptace, Pravděpodobnostní lineární diskriminační analýza, PLDA, I-vektor.

Keywords

Recognition, speaker recognition, clustering, adaptation, Probabilistic Linear Discriminant Analysis, PLDA, I-vector.

Citace

Ondřej Novotný: Adaptace systémů pro rozpoznání mluvčího, diplomová práce, Brno, FIT VUT v Brně, 2014

Adaptace systémů pro rozpoznání mluvčího

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Oldřicha Plchota.

.....
Ondřej Novotný
26. května 2014

Poděkování

Chtěl bych poděkovat svému vedoucímu Oldřichu Plchotovi za odborné vedení, pomoc a cenné rady při vývoji této práce a pochopení dané problematiky. Dále chci poděkovat rodině a přítelkyni za jejich nejen morální podporu při tomto studiu.

Obsah

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Rozpoznávání řeči a mluvčích je dnes velmi žádaným a zkoumaným oborem. Systémy na rozpoznávání jsou široce využívány v různých oblastech lidského života. Své uplatnění našly v civilní, policejní i vojenské sféře. Od autorizace mluvčích, až k forenznímu dokazování. Na tyto systémy jsou neustále kladeny vyšší nároky. Požadavky na kvalitu signálu se snižují, kdežto požadavky na přesnost systémů se neustále zvyšují. Je nutné tyto systémy neustále přizpůsobovat stále novým podmínkám. Tomuto problému lze čelit návrhem nových metod pro nové podmínky, či hledáním způsobů jak fungující systém, těmto podmínkám přizpůsobit . A právě tato problematika bude náplní této práce. Konkrétně se zaměříme na možnosti adaptace systémů rozpoznávání řečníků na nové podmínky. Bude se jednat o přizpůsobení již existujících a fungujících systémů na nový typ nahrávek (novou doménu).

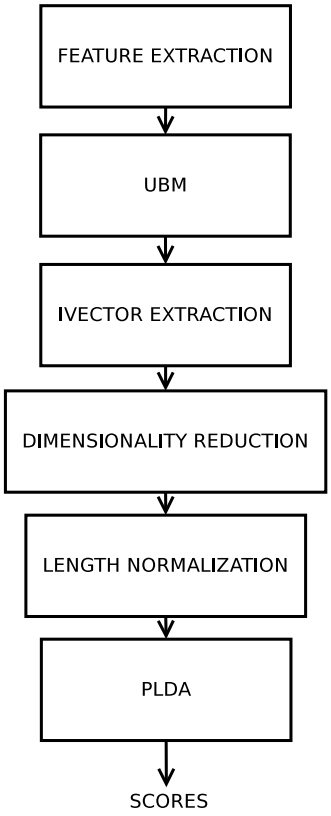
V této problematice existuje několik možných přístupů, podle dostupných dat, jenž můžeme k adaptaci použít. Důležitá je zde dostupnost anotací s identitou mluvčích u jednotlivých nahrávek. V této práci se budeme soustředit na nejpřísnější možnou situaci, kdy k samotné adaptaci máme pouze určité množství neanotovaných dat. Teoreticky se zaměříme na několik možností, jenž lze v adaptaci použít, které mezi sebou porovnáme.

V jednotlivých kapitolách se zaměříme postupně na aspekty s touto problematikou souvisejících. Nejdříve se zaměříme na použitý systém na rozpoznávání mluvčích a možnosti jeho adaptace. Následovat bude přehled shlukovacích technik, jejich vyhodnocení a možnosti odhadnutí počtu shluků. Dále se budeme zabývat zvýšením přesnosti shlukování a způsobu měření chyby systému. V závěru vyhodnotíme jednotlivé kroky adaptace podle celkového vlivu na chybu systému. Na konec porovnáme úspěšnost jednotlivých adaptačních technik v reálných podmínkách a provedeme klíčové srovnání mezi adaptací s anotovanými a neanotovanými nahrávkami.

Kapitola 2

Systém na rozpoznávání řečníka

Systém pro který je adaptace vyvíjena a na kterém je testována je systém vyvíjený a používaný ve výzkumné skupině **Speech@FIT** Ústavu počítačové grafiky a multimédií.



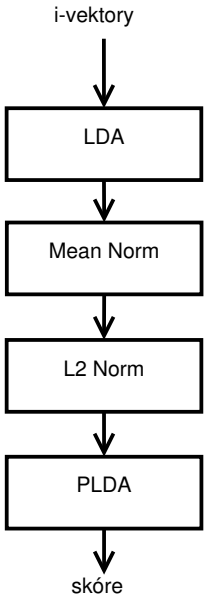
Obrázek 2.1: Zjednodušené schéma systému rozpoznávání mluvčích.

Na obrázku 2.1, lze vidět zjednodušené schéma běhu systému (viz [14]). Krátce si jej nyní popíšeme a prodiskutujeme použité parametry. V prvé řadě se budeme soustředit na části klíčové k tématu této práce.

V extrakci příznaků (**Feature Extraction**) slouží jako příznaky jednotlivých nahrávek *mel-frekvenční kepstrální koeficienty* [23], energie a delta a double-delta koeficienty. Příznaky jsou vypočítávány z nahrávek rozdělených na $20ms$ okna s $10ms$ překryvem.

Univerzální model pozadí (**UBM** - Universal background model) slouží jako model reprezentující rozložení příznaků nezávisle na mluvčím (viz [17]). Tento model je v plném nasazení tvořen 2048 gaussovkami. K tomuto kroku nejsou nutné anotace identifikující mluvčí jednotlivých nahrávek. Nahrávek je potřeba velké množství. Trénování bez anotací budeme označovat jako trénování bez učitele (**unsupervised**).

Dalším krokem je generování i-vektorů pro jednotlivé nahrávky. I-vektor je nízko dimenzionální vektor konstantní délky nezávisle na délce nahrávky (viz [6]). V základním nastavení je rozměr generovaných i-vektorů roven 600. Podrobně je cesta zpracování i-vektoru nastíněna na snímku 2.2. K redukci dimenzionality je využita Lineární diskriminační analýza (**LDA** - Linear discriminant analysis, viz [3]), která má za cíl nalézt podprostor, kde jsou třídy (mluvčí) nejlépe separovatelné (zde tedy již potřebujeme anotace k jednotlivým i-vektorům). Trénování s anotacemi se označuje jako učení s učitelem (**supervised**). I-vektory jsou poté normalizovány pomocí střední hodnoty všech i-vektorů (*mean-norm*) takto: $\mathbf{x}_n = \mathbf{x} - \boldsymbol{\mu}$ ($\mathbf{x}_n$,$\mathbf{x}$ označují normovaný a nenormovaný vektor, $\boldsymbol{\mu}$ střední hodnotu všech vektorů). Dále jsou i-vektory normovány na jednotkovou délku: $\mathbf{x}_n = \frac{\mathbf{x}}{||\mathbf{x}||_2}$, $||\mathbf{x}||_2 = \sqrt{\sum_{r=1}^{N} |x_r|^2}$, kde x_r jsou jednotlivé prvky vektoru $\mathbf{x}$ (viz [12]).



Obrázek 2.2: Schéma zpracování i-vektorů v systému.

Po normalizaci získáváme i-vektory ležící na povrchu hyperkoule.

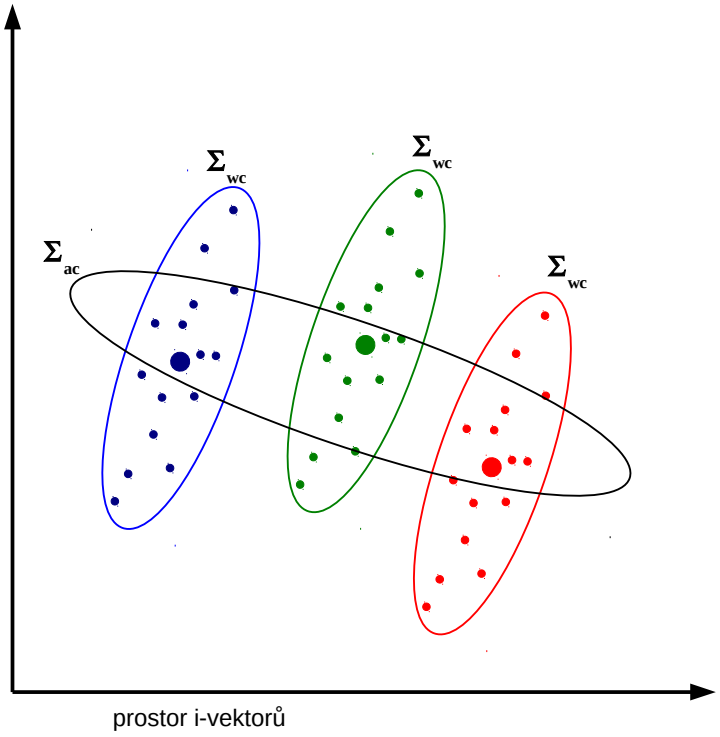
Posledním krokem je trénování a skórování i-vektorů pomocí **PLDA** (Probabilistic Linear Discriminant Analysis) systému. Tento krok opět vyžaduje ke svému natrénování, znalost anotací identifikující jednotlivé nahrávky. Tento krok bude podrobněji popsán dále v textu, neboť právě na něj se bude při adaptaci nejvíce soustředit naše úsilí.

2.1 Pravděpodobnostní lineární diskriminační analýza

Pravděpodobnostní lineární diskriminační analýza (dále jen PLDA) je metoda založená na sdružené faktorové analýze. V úlohách pro rozpoznání mluvčích je užívána ke klasifikaci i-

vektorů. Stejně je tomu tak i v systému použitém v této práci. Tato část vychází z [16],[20] a [11].

Podívejme se na speciální typ PLDA, kterým je dvoukovarianční model (popsán v článku [5] a přehledněji v práci [11]). Jedná se o model ve kterém se variabilita mluvčích a variabilita kanálu předpokládá v gausovském rozložení. Variability jsou poté reprezentovány mezitřídní (*across-class*) a třídní (*within-class*) kovarianční maticí $\mathbf{\Sigma}_{ac}$ a $\mathbf{\Sigma}_{wc}$ (viz 2.3).



Obrázek 2.3: Ukázka prostoru i-vektorů a jejich rozložení. Tučné body odpovídají identitě mluvčích a jejich rozložení odpovídá kovarianční matici $\mathbf{\Sigma}_{ac}$ (variabilita mluvčích). Rozložení i-vektorů známých mluvčích odpovídá $\mathbf{\Sigma}_{wc}$ (variabilita kanálu).

Předpokládaná pravděpodobnost identity mluvčího $\mathbf{y}$ odpovídá 2.1. Následně distribuce i-vektorů $\boldsymbol{\phi}$ známého mluvčího $\hat{\mathbf{y}}$ odpovídá 2.2. Proměnná $\boldsymbol{\mu}$ reprezentuje střední hodnotu identit mluvčích.

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \mathbf{\Sigma}_{ac}) \tag{2.1}$$

$$p(\boldsymbol{\phi}|\hat{\mathbf{y}}) = \mathcal{N}(\boldsymbol{\phi}; \hat{\mathbf{y}}, \mathbf{\Sigma}_{wc}) \tag{2.2}$$

Obecný model PLDA je spíše podobný Sdružené faktorové analýze (**JFA** - Joint Factor Analysis). I-vektrory mohou být dekomponovány jako:

$$\boldsymbol{\phi} = \boldsymbol{\mu} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x} + \boldsymbol{\epsilon} \tag{2.3}$$

Model lze rozdělit do dvou částí, $\boldsymbol{\mu} + \mathbf{Vy}$ popisuje prostor mluvčího (neměnná pro všechny nahrávky daného mluvčího). Proměnná $\mathbf{y}$ je skrytá proměnná reprezentující mluvčího. Část $\mathbf{Ux} + \boldsymbol{\epsilon}$ reprezentuje kanál a akustické podmínky specifické pro každou nahrávku. Proměnná $\mathbf{x}$ je skrytá proměnná reprezentující kanál. Proměnná $\boldsymbol{\epsilon}$ reprezentuje residuální variabilitu. Matice $\mathbf{V}$ a $\mathbf{U}$ reprezentují dané podprostory (mluvčího a kanálu).

Dříve zmíněné třídní a mezi-třídní kovarianční matice, jsou dány vztahy $\boldsymbol{\Sigma}_{ac} = \mathbf{VV}'$ a $\boldsymbol{\Sigma}_{wc} = \mathbf{UU}'$. Více se nebudeme PLDA věnovat, z hlediska adaptací je pro nás zmíněná teorie dostačující. Pro studium trénování hyperparametrů odkážeme na uvedenou literaturu.

Při vyhodnocování podobnosti dvou i-vektorů je skóre definováno jako poměr logaritmických věrohodností mezi dvěma hypotézami $\mathcal{H}_1$ a $\mathcal{H}_2$. Takovými, že i-vektory $\boldsymbol{\phi}_1$ a $\boldsymbol{\phi}_2$ jsou od stejného mluvčího (hypotéza $\mathcal{H}_1$), nebo ne ($\mathcal{H}_2$). Skóre $s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \log(p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathcal{H}_1)/p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathcal{H}_2))$. Podrobný výpočet skóre viz [11] a [16].

Kapitola 3

Možnosti adaptace systému

Dostáváme se ke klíčové pasáži této práce. Nyní se budeme zabývat možnostmi adaptace zmíněného systému v kapitole 2.

Systémy na rozpoznávání mluvčích jsou vždy trénovány na určitých nahrávkách. Tyto nahrávky mívají vždy něco společné. Ať jsou tyto prvky lidskému uchu slyšitelné, či nikoli, mají neblahý efekt na přesnost rozpoznávání nahrávek nových. V případě, že tyto nové nahrávky nesdílejí tyto společné znaky, mají tyto nahrávky jinou variabilitu, než kterou očekává systém a chyba rozpoznávání tedy roste. Je proto nutné systém nějakým způsobem přizpůsobit, tedy adaptovat novým nahrávkám.

Klíčovým úkolem adaptace je tedy přizpůsobení systému na novou doménu. Pod pojmem domény si můžeme představit kanál přenosu nahrávky (šum, akustické podmínky), či jazyk nahrávek atd. Zkrátka se jedná o proměnné, které mají vliv na rozpoznávání mluvčích a jejich změna mezi trénovací a evaluační sadou vede ke snížení přesnosti rozpoznávání.

Na základě námi zkoumaného systému nyní probereme možnosti adaptace a potřebné podmínky, které pro užití zmíněných metod musíme splnit. Adaptaci lze rozdělit do dvou hlavních větví. Na adaptaci s učitelem a na adaptaci bez učitele. Podle toho zda máme k dispozici u adaptačních dat anotace, či nikoli. Prozkoumáme možnosti z obou těchto větví.

Pro zjednodušení úkolu nebudeme ve zmíněném systému uvažovat dimenzionální redukci, která taktéž vyžaduje u nahrávek anotace, které u většiny adaptačních technik nebudeme mít k dispozici. Budeme tedy uvažovat i-vektory v plné velikosti 600. Z tohoto kroku vyplývá, že následující adaptace se budou soustředit pouze na skórování pomocí **PLDA**.

3.1 Nově natrénovaný systém

Nejedná se přímo o adaptační techniku. Bude ovšem dobré tuto možnost zmínit, k porovnání vůči ostatním technikám a tedy ke zjištění důvodu, proč se k adaptaci uchylujeme, místo trénování nového systému. Jedná se o nové natrénování systému pro nahrávky z nové domény. Krok se to může zdát jednoduchý, vyžaduje ovšem nemalé zdroje. Je totiž nutné mít velké množství trénovacích nahrávek a k nim patřičné anotace pro jejich identifikaci. Tato podmínka může být v praxi velmi těžko splnitelná, případně i nemožná. Jestliže disponujeme takovouto databází nahrávek, jedná se o nejvhodnější krok, neboť tímto krokem dosáhneme nejmenší možné chybovosti.

3.2 Adaptace s učitelem

V tomto kroku je taktéž nutné mít nahrávky s anotacemi. Konkrétně nám stačí pouze i-vektory vygenerované neadaptovaným systémem. Pomocí těchto i-vektorů a jim příslušným anotacím přetrénujeme pouze blok se skórováním, tedy **PLDA**. Z hlediska chybovosti nedosáhneme zřejmě takových výsledků, jako v nově natrénovaném systému. Dosáhneme ovšem nejlepších výsledků možných v adaptaci. Tyto výsledky nám poté budou vymezovat interval, v kterém se budeme při adaptacích pohybovat. Nevýhoda této techniky je zřejmá, opět je nutné mít klíče k jednotlivým nahrávkám.

3.3 Adaptace bez učitele

Dostáváme se k jádru adaptačních technik. Jsou to techniky, kdy u adaptačních nahrávek nemámě žádné anotace. V praxi se bude jednat asi o nejčastější situaci. Získání nahrávek může být snadné, ale jejich anotace už by vyžadovala jistě další prostředky, což nemusí být vždy v reálných možnostech provozovatele systému.

V této práci zmíníme několik takových technik. Techniky rozdělíme do tří skupin: **Shlukování a přetrénování**, **Shlukování a adaptace** (viz [10]) a **Podvrhnutí PLDA modelu** (viz [22]).

3.3.1 Shlukování a přetrénování

Princip spočívá v použití neadaptovaného systému PLDA k výpočtu skóre adaptačních i-vektorů (jejich podobnost). Následuje odhad identifikace mluvčích jednotlivých i-vektorů pomocí některé ze shlukovacích technik. Na základě provedeného odhadu identifikace je natrénován nový PLDA model pomocí adaptačních dat na novou doménu. Obecná metoda je nastíněna v pseudokódu 1.

Algoritmus 1: Pseudokód jednoduché adaptace s přetrénováním systému.

input : PLDA model
output: adaptovaný PLDA model

for $i \leftarrow 1$ **to** N **do**
    score = `scoring` (PLDA,adapt_data);
    labels = `clustering` (score);
    train_data = adapt_data + η data;
    PLDA = `train` (train_data,labels);
end

Proměnná η určuje podíl použitých mimo-doménových trénovacích vektorů. Množství iterací bude u algoritmu záviset na rychlosti konvergence chyby rozpoznávání ke stabilní nižší hodnotě. Vhodná hranice ukončení iterací by mohla být určena změnou skóre adaptačních dat (velikost změny pod určitým prahem). Povšimněme si, že pro natrénování nového PLDA systému nevyužíváme pouze vstupních adaptačních dat, ale i mimo-doménových dat, původně k natrénování mimo-doménového PLDA systému. K tomuto spojení trénovacích dat se uchylujeme, neboť přesnost odhadu identity mluvčích nemusí být po prvním kroku natolik dostatečná, aby v dalších iteracích vedla ke konvergenci řešení s menší chybou.

Při spojování trénovacích množin poté máme dvě možnosti:

1. Nechat podíl mimo-doménových dat určených k tréningu konstantní.

2. Postupně podíl mimo-doménových dat snižovat.

Zprvu nám mohou mimo-doménová trénovací data poskytnout podporu pro snížení chyby při trénování nového PLDA modelu, zároveň tímto spojením zůstává v trénovací sadě nechtěná mimo-doménová variabilita dat (viz $\mathbf{\Sigma}_{ac}$ a $\mathbf{\Sigma}_{wc}$ v 2.1). Proto by mohlo být dobré podíl těchto dat postupně snižovat.

Metoda dále vychází z předpokladu, že máme k dispozici mimo-doménovou trénovací sadu.

3.3.2 Shlukování a adaptace

Opět využíváme neadaptovaný mimo-doménový PLDA systém k ohodnocení podobnosti adaptačních i-vektorů a následnému odhadu identifikace mluvčích.

Na základě odhadnuté identifikace natrénujeme nový PLDA systém v adaptované doméně. Pracujeme s parametry modelu PLDA systému $\mathbf{\Sigma}_{ac}$ a $\mathbf{\Sigma}_{wc}$. Výsledný PLDA systém je dán váženým průměrem doménového a mimo-doménového systému PLDA viz 3.1. Indexy $_{in}$ a $_{out}$ označují doménové a mimo-doménové matice ([10]).

$$(\mathbf{\Sigma}_{ac}, \mathbf{\Sigma}_{wc}) = arg\max \alpha(\mathbf{\Sigma}_{ac}, \mathbf{\Sigma}_{wc})_{in} + (1-\alpha)(\mathbf{\Sigma}_{ac}, \mathbf{\Sigma}_{wc})_{out} \tag{3.1}$$

Parametr α ,kde $0 < \alpha < 1$, lze použít jednotný pro obě matice $\mathbf{\Sigma}_{ac}$ a $\mathbf{\Sigma}_{wc}$. Vhodnějším krokem ovšem bude použít rozdílné parametry pro obě kovarianční matice. Označme tyto parametry α_{wc} a α_{ac} ($0 \leq \alpha_{wc} \leq 1$ a $0 \leq \alpha_{ac} \leq 1$) pro kovarianční matice $\mathbf{\Sigma}_{wc}$ a $\mathbf{\Sigma}_{ac}$.

Rozdílná nastavení těchto parametrů by měla odpovídat následujícímu:

- α_{ac} - Velikost by měla odpovídat poměru množství mluvčích použitých k trénování mimo-doménového a doménového systému PLDA. Dále by měla pokrývat rozdíl mezi parametry mluvčích na doménových a mimo-doménových datech.

- α_{wc} - Velikost by měla především pokrýt míru rozdílnosti přenosových kanálů u doménových a mimo-doménových dat.

Zmíněnými parametry mluvčích máme namysli například jazyk. Z uvedené možnosti nastavení rozdílných parametrů α_{wc} a α_{ac} nám vyplývá možnost měnit při adaptaci pouze potřebnou variabilitu.

Zamysleme se nad modelovou situací, kdy máme doménová i mimo-doménová data obsahující stejné mluvčí a tedy i stejnou $\mathbf{\Sigma}_{ac}$ variabilitu. Akustické podmínky obou typů nahrávek budou ovšem rozdílné. V takovémto případě by mělo postačovat adaptovat pouze rozdílnou variabilitu $\mathbf{\Sigma}_{wc}$ a variabilitu $\mathbf{\Sigma}_{ac}$ použít z původního PLDA systému.

Po vytvoření nového modelu PLDA daného váženým průměrem, máme možnost použít tento systém přímo ke skórování, či jej použít jako inicializaci a provést na něm ještě několik trénovacích iterací. Adaptační funkce je uvedena v pseudokódu 2.

Systém této metody se systémem předchozí se může zdát stejný. Stejný by byl pouze v případě, kdy by parametr η z předchozí byl roven 0 a parametry α z této metody rovny 1. Jde o případ, kdy se využije v adaptovaném systému pouze v adaptační datové sady. V případě jiných hodnot parametrů se systémy již liší. Předchozí metoda poté provádí rozšíření mimo-doménové variability (mimo-doménová trénovací sada PLDA) pomocí adaptační sady. Naproti tomu tady zmíněná metoda provádí pomocí váženého průměru kovariančních matic, postupný přechod od jedné variability k druhé.

Algoritmus 2: Pseudokód adaptace s váženým průměrem systémů.

input : PLDA model
output: adaptovaný PLDA model

for $i \leftarrow 1$ **to** N **do**
    score = `scoring` (PLDA,adapt_data);
    labels = `clustering` (score);
    PLDA_{new} = `train` (adapt_data,labels);
    PLDA.WC = α_{wc} $\text{PLDA}_{new}.WC + (1 - \alpha_{wc})$PLDA.WC
    PLDA.AC = α_{ac} $\text{PLDA}_{new}.AC + (1 - \alpha_{ac})$PLDA.AC
end

3.3.3 Podvrhnutí PLDA modelu

Třetí možnost popsaná v [22] je důsledkem zvážení, zda je vůbec nutné používat nějaký shlukovací algoritmus. Za předpokladu, že budeme provádět adaptaci pouze ve variabilitě mluvčích ($\mathbf{\Sigma}_{ac}$ nebo $\mathbf{V}$ v závislosti na PLDA). V takovém případě by bylo možné odhadnout tuto variabilitu na základě všech adaptačních i-vektorů jako jejich kovarianci. Nové parametry budou v tomto případě dány viz 3.2. Množina adaptačních i-vektorů je označena $\mathbf{X}$, α se opět pohybuje v intervalu $\langle 0, 1 \rangle$.

$$\mathbf{\Sigma}_{ac} = \alpha \mathbf{\Sigma}_{ac} + (1 - \alpha) Cov(\mathbf{X}) \tag{3.2}$$

Tento způsob adaptace bude ovšem vhodné použít pouze v okamžiku, kdy nám na jednoho mluvčího připadá v adaptační sadě pouze jeden i-vektor (jedna nahrávka). V takovémto případě, lze vypočtenou kovarianci považovat za odhad kovariance mluvčích.

V případě, kdy v adaptačních datech bude připadat jednomu mluvčímu více i-vektorů se ovšem dopouštíme chyby. Za těchto podmínek se snažíme adaptovat variabilitu mluvčích pomocí, totální kovariance všech dat $\mathbf{\Sigma}_t$, která jistě kovarianci mluvčích neodpovídá. V těchto případech poté $\mathbf{\Sigma}_t$ spíše odpovídá (pokud se omezíme na samotné LDA) součtu kovariancí $\mathbf{\Sigma}_{ac}$ a $\mathbf{\Sigma}_{wc}$, tedy $\mathbf{\Sigma}_t = \mathbf{\Sigma}_{ac} + \mathbf{\Sigma}_{wc}$ (viz LDA v [20]).

Kapitola 4

Shlukovací metody a odhad počtu shluků

V této kapitole probereme jednotlivé algoritmy pro shlukování a odhad počtu shluků, které byly použity při vývoji a testech.

Z počátku bude dobré si uvědomit, na základě jakých dat v naší problematice můžeme toto shlukování provádět. K dispozici máme dvě možnosti. Použít přímo adaptační i-vektory jako body prostoru a pomocí nich a vhodných technik provádět odhad klíčů (mluvčích). Výhodou tohoto přístupu je znalost absolutní polohy dat.

Druhou možností je využití vypočítaného skóre (neadaptovaného) systému jako vzdálenosti mezi jednotlivými vektory. Přístup si lze představit jako topologickou mapu s ohodnocenými hranami. Pomocí technik shlukování grafových struktur budeme opět provádět odhad identifikace mluvčích. Výhodou tohoto přístupu budou možné iterace se zlepšujícím se skóre. Nevýhodou je, že nemáme informaci o absolutní poloze bodů v prostoru, což může mít negativní dopad na úspěšnost některých shlukovacích algoritmů.

4.1 Vzdálenost a podobnost vektorů

Nyní si vymezíme pojmy, které nás budou provázet celou kapitolou.

Měřením vzdálenosti myslíme měření vzdálenosti mezi dvěma vektory (body určenými těmito vektory) v metrickém prostoru. V praxi se k měření užívá několik odlišných vzdáleností, kdy typ vzdálenosti volíme v závislosti na povaze dané úlohy. Zde se omezíme na vzdálenost Euklidovu: $d_e(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_i (p_i - q_i)^2}$ a kosinovou: $d_c(\mathbf{p}, \mathbf{q}) = 1 - \frac{\mathbf{p} \cdot \mathbf{q}}{\sqrt{(\mathbf{p} \cdot \mathbf{p})(\mathbf{q} \cdot \mathbf{q})}}$, kde $\mathbf{p}$ a $\mathbf{q}$ značí vektory, p_i a q_i pak hodnoty vektorů na pozici i (viz [21]).

Shlukovací algoritmy využité při odhadu identit nebo i počtu shluků v této práci, pracují na dvou rozdílných principech (využívajících různé metriky) řazení vektorů do skupin. První metrikou je vzdálenost, která nulovou hodnotou označuje stejné vektory. Druhá metrika je podobnost, jenž v ideálním případě je obrácená hodnota vzdálenosti, podle zvolené maximální hodnoty. Zvolená maximální hodnota poté reprezentuje situaci, kdy se jedná o stejné vektory. S přepočtem mezi vzdáleností a podobností se setkáváme již ve zmíněné kosinové vzdálenosti, kdy ve vzorci od hodnoty 1 odečítáme podobnost. Hodnota 1 je maximum jakého může hodnota podobnosti dosáhnout a reprezentuje shodnost vektorů. Z uvedené logiky vyplývá, že podobnost i vzdálenost je vždy z jedné strany ohraničena (0, či maximální podobností). Zároveň máme jistotu, že při měření vzdálenosti identických vektorů dostaneme vždy konstantní hodnotu (př. 0 u vzdálenosti).

Z nastíněných adaptačních technik v kapitole 3, chceme v adaptaci iterovat za účelem snižující se chyby. Z tohoto důvodu nelze použít vektory k výpočtu vzdálenosti (podobnosti), byla by konstantní, proto tedy využíváme skóre PLDA jako *virtuální podobnost* a pomocí tohoto skóre vypočítáváme v případě potřeby i *virtuální vzdálenost*. Tyto pojmy budeme dále používat v souvislosti shlukování s využitím skóre PLDA.

PLDA skóre použité jako podobnost není obecně ani z jedné strany ohraničené hodnotou, kterou by nepřekročilo (rozsah hodnot se u různě natrénovaných systémů navíc liší). S tímto souvisí i nejednotnost hodnoty virtuální podobnosti vektorů se sebou samých, tedy různé vektory jsou sobě samým různě podobné.

Běžně by jsme použili konstantní hodnotu podobnosti stejných vektorů k výpočtu vzdálenosti: $d = S_{konst} - s$, kde S_{konst} je hodnota podobnosti stejných vektorů, s jedna daná hodnota podobnosti a d výsledná vzdálenost. Jelikož nám tato konstantní hodnota chybí, musíme přepočet upravit. Podobnost na vzdálenost (a opačně) transformujeme tedy jednoduchým vztahem: $d = max(\mathbf{S}) - s$, kde $\mathbf{S}$ je množina všech podobností, které chceme transformovat, s jedna daná podobnost a d příslušná vzdálenost.

4.2 Vyhodnocení kvality shlukování

V prvé řadě uveďme evaluační metriky, které nám budou sloužit k porovnání jednotlivých shlukovacích technik.

Pro hodnocení kvality shlukovacích metod, tedy to jak přesně budou odhadnuté shluky odpovídat skutečným třídám, byly vybrány dvě metriky. První metrikou je čistota shluku (v angličtině označována jako *cluster purity*). Druhou metrikou je f-skore (v angličtině označována jako *f1-score*, *f-score*, či *f-measure*).

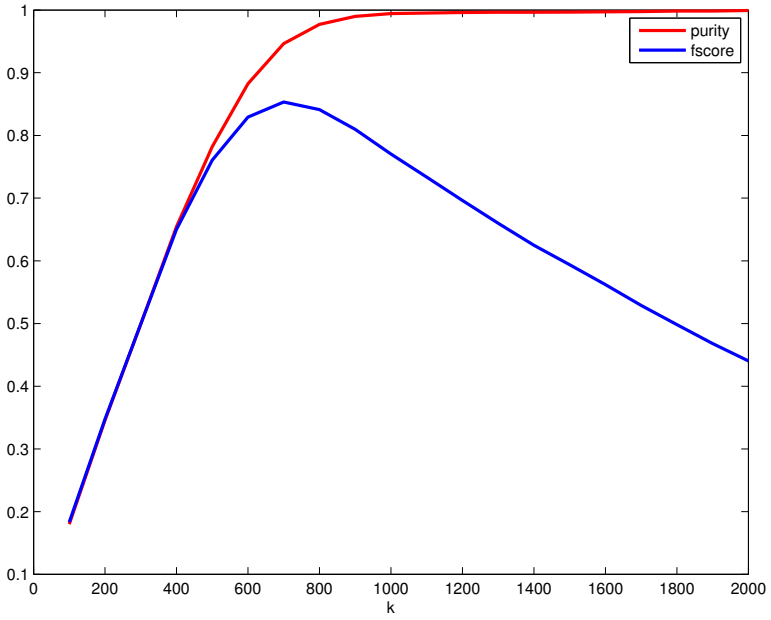
4.2.1 Čistota shluku

Tato metrika je pro výpočet velmi prostá a přesto dosti efektivní. Obor hodnot této metriky je v rozsahu $(0, 1\rangle$. Jednička odpovídá naprosto přesnému určení shluků, třídám skutečným. Výpočet této metriky je pro odhadnuté shluky Ω a skutečné třídy C uveden v rovnici 4.1, ω_k a c_j odpovídají množinám prvků v k-tém odhadnutém shluku a j-té třídě (N je celkový počet prvků). Podrobnější popis k této evaluační metrice a dalším lze nalézt v knize [13].

$$p(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \tag{4.1}$$

Pro jednotlivé odhadnuté shluky se podle skutečných tříd najde dominantní třída v daném shluku a určí počet dat. Čistota shluku je poté dána podílem součtu množství dominantních tříd ve shlucích s celkovým počtem dat.

Jak je vidět, princip není nijak složitý, ale z popsaného algoritmu je zřejmá zásadní nevýhoda daného algoritmu. Algoritmus bude dávat nejpřesnější výsledky ohledně přesnosti shlukování v případě stejného (nebo menšího) počtu shluků, jako skutečných tříd. V případě nižšího počtu shluků budou výsledky ještě použitelné, kdežto v případě vyššího počtu shluků, dochází k nepříznivému efektu a to, že v případě chybného shlukování ukazují výsledky stále kvalitní metodu. Metoda tedy nijak neřeší chybné určení počtu shluků.



Obrázek 4.1: Ukázka vývoje čistoty shluků a f-skóre v závislosti na počtu shluků (správným počtem bylo 600).

4.2.2 *f*-skóre

Oproti Čistotě shluku má tato metrika již složitější postup výpočtu. Její výhoda je ovšem v usměrnění výsledku i vůči chybnému určení počtu shluků. Výsledky této metody opět spadají do rozsahu $(0, 1\rangle$. Při výpočtu f-skóre se využívá dvou pomocných metrik označovaných jako **Precision** $P(C_r, \Omega_i) = \frac{n_{ri}}{N_i}$ a **Recall** $R(C_r, \Omega_i) = \frac{n_{ri}}{N_r}$. Význam těchto metrik a jejich možné další použití je podrobněji popsáno v knize [18]. C_r označuje r-tou třídu prvku o velikosti N_r, Ω_i označuje shluk o velikosti N_i a n_{ri} označuje počet prvků ve shluku Ω_i ze třídy C_r. Vycházíme především z materiálů [18] a [24]. Při výpočtu se počítá pomocné f-skóre mezi všemi dvojicemi shluků a tříd (viz 4.2). Na tomto základě se určí f-skóre každé třídě (viz 4.3). Výsledné f-skóre celého řešení je dáno aplikací 4.4.

$$F(C_r, \Omega_i) = 2 \cdot \frac{P(C_r, \Omega_i) \cdot R(C_r, \Omega_i)}{P(C_r, \Omega_i) + R(C_r, \Omega_i)} \tag{4.2}$$

$$F(C_r) = \max_{\Omega_i} \left(F(C_r, \Omega_i)\right) \tag{4.3}$$

$$FScore = \sum_{r=1}^{R} \frac{N_i}{N} F(C_r) \tag{4.4}$$

Výhoda této evaluační metriky spočívá v tom, že postihuje i správnost odhadnutého počtu shluků. Doplní metriku *Čistota shluků* především v oblastech, kdy odhadnutý počet shluků přesáhne skutečný počet tříd. Porovnání obou metrik lze vidět na grafu 4.1.

4.3 Shlukovací metody

4.3.1 Aglomerativní hierarchické shlukování

V případě této shlukovací techniky budeme vycházet z [26] a [8]. Aglomerativní hierarchické shlukování (dále jen AHC) je shlukovací metoda, využívající pro shlukování vzdálenost mezi prvky. V problematice adaptace zde využijeme virtuální vzdálenost.

Nebudeme zde uvádět algoritmus, podrobný popis lze nalézt v [8]. V počátečním stavu algoritmu jsou shluky tvořeny jednotlivými daty. V každé iteraci tohoto algoritmu jsou spojeny shluky, které mají mezi sebou nejmenší vzdálenost. Tento krok se opakuje, dokud nezískáme jediný shluk, či nedosáhneme požadovaného množství shluků. Výstupem algoritmu je taktéž hierarchická stromová struktura odpovídající jednotlivým spojením. Hloubkou zanoření v této struktuře lze poté regulovat počet shluků.

Zmíněná možnost vygenerovat hierarchickou strukturu až do jediného shluku nemá příliš velký význam z hlediska samotné klasifikace. Je ovšem výhodná v technikách, kdy budeme provádět odhad počtu shluků. Lze totiž provést pouze jedno shlukování a posléze si jen vybírat množství shluků podle hloubky zanoření a tím i patřičnou identifikaci mluvčích. Z toho plyne urychlení výpočtů při postupném testování jednotlivých počtů shluků a jejich následném vyhodnocení.

4.3.2 Spojování komponent

Opět se jedná o shlukování v grafové struktuře váhovaného grafu. Námět k metodě je převzat z [22].

Ohodnocení hran zde bude odpovídat podobnosti dvou prvků, případně virtuální podobnosti. Princip je velice jednoduchý. Lze jej přirovnat k metodě postupného zaplavování, kdy zvedáme hodnotu prahu. Hrany s nižší hodnotou než je daný práh se odstraňují. Zbylé prvky se pospojují. Vznikne tak několik propojených shluků.

Drobné úskalí vzniká v okamžiku, kdy potřebujeme najít takovou hodnotu prahu, která bude odpovídat požadovanému množství shluků. Zde nemá smysl začínat s prahem od nuly a postupně ho zvyšovat o předem stanovený krok. Tímto způsobem by se požadovaný práh mohl snadno minout, zvláště při větší velikosti kroku. Na druhou stranu příliš malý krok povede k extrémní časové náročnosti. Vhodný kompromis pro hledání optimálního prahu nám nabízí *Metoda půlení intervalů*, která nám nabízí logaritmickou časovou složitost.

4.3.3 Markovův shlukovací algoritmus

V základní verzi ([7]) tato metoda využívá grafu s neváhovanými hranami. Vychází se z předpokladu, že mezi prvky téhož shluku bude více hran, než mezi prvky jednotlivých shluků. Lze to aplikovat do problematiky váhovaných grafů, kde mezi prvky téhož shluku jsou propojovací cesty kratší, než mezi prvky shluků opačných.

Jako reprezentace grafu je v algoritmu využita matice spojení. Algoritmus simuluje náhodný průchod grafem pomocí střídání operací nazývaných **expanze** a **inflace**.

Jako expanzi uvažujeme n-tou mocninu matice grafu. Dilatace $(\Gamma_r \mathbf{M})_{pq}$ matice $\mathbf{M} \in \mathbb{R}^{k \times l}$ pro $r > 1$je definována viz 4.5.

$$(\Gamma_r \mathbf{M})_{pq} = \frac{(M_{pq})^r}{\sum_{i=1}^{k} (M_{iq})^r} \tag{4.5}$$

Před samotným krokováním algoritmu je nutné vytvořit Markovskou reprezentaci grafu (někdy označovanou též jako normalizace). Jedná se o přepočet grafu s váhami na graf s hranami ohodnocenými pravděpodobností přechodu z daného uzlu do uzlu sousedního. Vyšší hodnota hrany tedy po normalizaci dává vyšší pravděpodobnost přechodu. Normalizace matice $\mathbf{M} \in \mathbb{R}^{k \times l}$ je následující 4.6. Normalizací ztrácí matice reprezentující graf symetričnost.

$$M_{pq} = \frac{M_{pq}}{\sum_{i=1}^{k} M_{iq}} \tag{4.6}$$

Nástin fungování samotného algoritmu je naznačen v pseudokódu 3.

Algoritmus 3: Pseudokód algoritmu MCL shlukování.

input : graf G,e,r
output: maticová reprezentace shluků T_k

```
G = G + Δ;                          /* přidání zpětné vazby každému prvku */
T_1 = T_G;                          /* tvorba Markovova grafu */
while T_k != T_{k-1} do
    T_k = (T_{k-1})^e;              /* expanze */
    T_k = Γ_r(T_k);                 /* inflace */
end
```

Výhoda tohoto algoritmu tkví v samostatném odhadu počtu shluků. Není tedy nutné takovou hodnotu hledat jiným algoritmem a vkládat ji jako vstupní parametr.

Druhou podstatnou výhodou tohoto algoritmu je výsledná tvorba shluků. Díky tomu, že je algoritmus založen na simulaci toku, není algoritmus zatížen na hledání shluků určitého rozložení. Tímto nedostatkem trpí většina algoritmů na odhad počtu shluků, dle nějaké vzdáleností funkce. Není totiž příliš objektivních charakteristik využívajících něčeho jiného, než je *střední čtvercová chyba*, která svou povahou nejvíce vyhovuje shlukovacímu algoritmu *k-means*, který se ji snaží při shlukování minimalizovat.

V základní verzi algoritmus nepředpokládá zpětnou vazbu prvků sama na sebe a je nutné ji v algoritmu doplňovat. V našem případě matici spojení reprezentuje matice skóre, kde již tato zpětná vazba je, a proto lze zmíněnou přípravu vynechat.

4.4 Kvalita shlukovacích algoritmů z pohledu adaptace

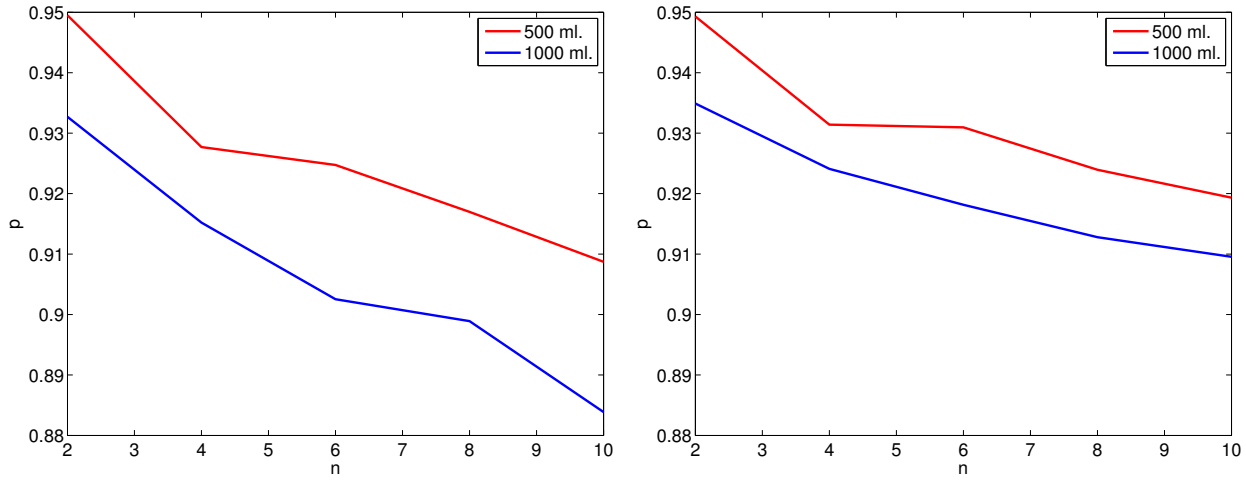
Probrali jsme několik shlukovacích metod, nyní analyzujme dosažené výsledky jednotlivých algoritmů v problematice adaptace.

Pro účel testování kvality jednotlivých algoritmů byly algoritmy testovány na dvou skupinách mluvčích. První skupina byla tvořena 500 náhodně vybranými mluvčími s různou, ale rovnoměrnou hustotou i-vektorů odpovídající jednotlivým mluvčím. Druhá skupina byla vybrána a nastavena obdobně, ovšem pro 1000 mluvčích.

Abychom se mohli soustředit přímo na kvalitu shlukování jednotlivých algoritmů, byla v této fázi testování algoritmů poskytnuta informace o množství shluků do kterého mají být i-vektory shlukovány. Vyhneme se tak nepřehledné chybovosti, která vzniká případným špatným odhadem počtu shluků.

Hned z počátku se vyskytl problém s konvergencí Markovova shlukování. Problém zřejmě spočívá v nastavení velikosti zpětnovazební smyčky jednotlivých uzlů. V našem případě, kdy používáme jako matici grafu matici se skóre z PLDA systému, jsou již smyčky obsaženy. Jelikož víme, že skóre na diagonále matice vždy odpovídají stejným i-vektorům, lze jej měnit, případně sjednotit. Jeho velikostí lze ovlivnit pak následnou pravděpodobnost přechodu z uzlu zpět do sebe samého. Byly vyzkoušeny dva přístupy. První, kdy bylo vložené skóre vysoké a následná pravděpodobnost tedy také, a druhý, kdy jsme zvolili střední cestu a nastavili tuto hodnotu na střední hodnotu všech skóre. Před tímto krokem bylo nutné všechna skóre posunout do kladných hodnot pomocí vztahu $s_n = s + |\min(S)| + 1$ (S reprezentuje množinu všech hodnot skóre, s a s_n původní hodnotu a posunutou hodnotu skóre). Přičtením jedničky zabráníme vzniku nulové pravděpodobnosti u nejmenších prvků. Vyzkoušeno bylo i úplné odstranění zpětné vazby, což není doporučováno, kvůli menší pravděpodobnosti konvergence. Výsledkem všech zmíněných úprav bylo dosaženo bohužel pouze dvou neuspokojivých výsledků. Prvním bylo řešení obsahující pouze jediný shluk pokrývající všechna vstupní data, a druhý, kdy množství nalezených shluků odpovídalo jednotlivým vstupním datům.

Přes tento neúspěch, v případě správného fungování, by tento algoritmus mohl poskytnou zajímavé výsledky. V případě, že ne z hlediska přesnosti shlukování, tak z hlediska odhadu množství mluvčích. Dále se v hodnocení shlukovacich metod omezíme na algoritmy *Spojování komponent* a *Aglomerativní hierarchické shlukování*.



Obrázek 4.2: Ukázka vývoje kvality hierarchického shlukování (vlevo) a spojování komponent (vpravo) na základě množství i-vekorů na mluvčího.

Jak je pomocí obrázků 4.2 vidět obě shlukovací metody vykazují zhruba stejnou přesnost. Přesto *Spojování komponent* viditelně vykazuje drobně lepší výsledky při nízké hustotě i-vektorů na mluvčího, tak i při hustotě vyšší. Podrobné výsledky dosažené pro obě shlukovací techniky lze nalézt v tabulce 4.1.

Předchozí měření kvality shlukování je lehce optimistické, neboť bylo prováděno za pomoci podobnosti ze systému natrénovaného na nahrávkách ze stejné domény. Tedy stavu, kdy opravdu víme, jak jsou si vektory podobné. V adaptaci ovšem nemáme z počátku takovou přesnost v podobnosti k dispozici. Klasifikaci tak provádíme v podstatě za pomoci

	Mluvčích	Čistota shluku	f-skóre
Spojování komponent	500	0.93	0.94
	1000	0.92	0.92
Aglomerativní h. shlukování	500	0.93	0.93
	1000	0.91	0.91
Markovův shlukovací alg.	–	nekonvergovalo k řešení	–

Tabulka 4.1: Srovnání shlukovacích algoritmů (podobnosti na základě systému trénovaného v doméně).

chybných údajů. Podívejme se tedy na výsledky shlukovacích algoritmů v případě podobnosti z mimo-doménového systému (obecně odpovídá stavu první iterace v adaptaci).

	Mluvčích	Čistota shluku	f-skóre
Spojování komponent	500	0.83	0.83
	1000	0.80	0.81
Aglomerativní h. shlukování	500	0.79	0.77
	1000	0.76	0.73

Tabulka 4.2: Srovnání shlukovacích algoritmů (podobnosti na základě systému trénovaného mimo doménu).

V tabulce 4.2 vidíme, že Spojování komponent si mimo správnou doménu vede podstatně lépe.

4.5 Odhad počtu shluků

V předchozích sekcích o shlukovacích algoritmech byly probrány i algoritmy, jenž vyžadují jako vstupní parametr počet shluků do kterého mají být data shlukována. Jelikož je nám i tento parametr neznámý, podívejme se na několik existujících metod odhadu počtu shluků.

U všech následujících metod používáme k odhadu počtu mluvčích shlukování podle podobnosti vektorů získaných od PLDA systému. Tato situace se může zdát jako matoucí, neboť shlukování neprovádíme na vzdálenosti pomocí které provádíme odhad počtu shluků (např. kosinová). Toto užití je ovšem výhodnější, neboť virtuální podobnost z PLDA systému je při shlukování daleko přesnější, než kosinová, či Euklidova vzdálenost. Zároveň nám, ale podobnost neumožňuje měřit vzdálenost bodů od středu shluků, což se při odhadu počtu často užívá. Uchýlili jsme se tedy k tomuto kombinovanému řešení.

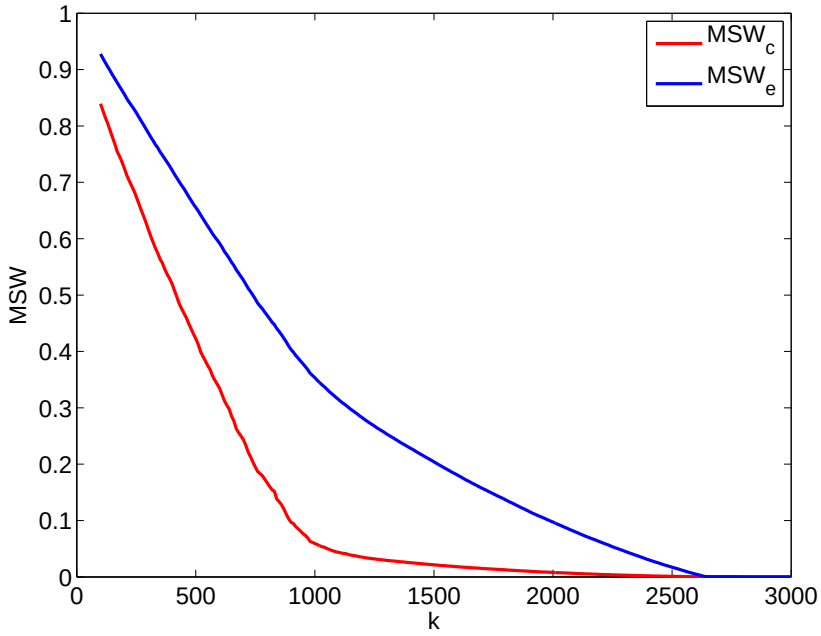
4.5.1 Odhad z průběhu křivky střední čtvercové vzdálenosti

Jak název napovídá, použijeme odhadu počtu shluků (mluvčích) vývoj křivky střední čtvercové vzdálenosti (označované jako MSW - *mean squares within the clusters*, či MSE - *mean square error* v případě Euklidovy vzdálenosti, viz [4] a [9]) mezi jednotlivými elementy a středy možných shluků. Střední čtvercová vzdálenost je definována jako 4.7 (s využitím Euklidovy vzdálenosti) a 4.8 (s využitím kosinové vzdálenosti), N odpovídá celkovému počtu datových prvků, x_i jednotlivým datovým prvkům a μ_{pi} střední odpovídajícího shluku.

$$MSW_e = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x_i} - \boldsymbol{\mu}_{pi}\|^2 \tag{4.7}$$

$$MSW_c = \frac{1}{N}\sum_{i=1}^{N} \left(1 - \frac{\mathbf{x}_i \cdot \boldsymbol{\mu}'_{pi}}{\sqrt{(\mathbf{x}_i \cdot \mathbf{x}'_i)(\boldsymbol{\mu}_{pi} \cdot \boldsymbol{\mu}'_{pi})}}\right) \tag{4.8}$$

V těchto technikách se využívá změny křivosti MSW hodnot v závislosti na počtu shluků, kdy se předpokládaná správná hodnota projeví jako výrazný ohyb v této křivce. Existuje několik způsobů jak toto řešení hledat. Z hlediska vzdálenosti mezi dvěma vektory pro výpočet MSW hodnoty se nám nabízí několik možností. V základu se při výpočtu této hodnoty uvažuje Euklidova vzdálenost, v dřívější době se ovšem pro vyhodnocení podobnosti i-vektorů používala kosinová vzdálenost. Byla do experimentů tedy také zařazena a ukázala se jako vhodnější (obr. 4.3).



Obrázek 4.3: Ukázka vývoje MSW křivek z Euklidovy a kosinové vzdálenosti. Z grafu je zřejmé, že kosinová vzdálenost lépe reflektuje hledané řešení k tedy 1000 shluků, než je tomu u Euklidovy vzdálenosti.

Při hledání optimálního počtu shluků je dále možno použít celkovou čtvercovou vzdálenost (SSW – *sum of squares within the clusters*, [9]) mezi jednotlivými elementy a středy možných shluků. Z hlediska samotného odhadu to nepřináší oproti MSW zásadní rozdíl.

Dále se při odhadu počtu shluků můžeme setkat s měřením celkové mezi shlukové vzdálenosti (SSB – *sum of square between the clusters*). Jedná se o celkovou vzdálenost mezi všemi kombinacemi shluků. SSB se nejčastěji využívá v kombinaci s MSW, kdy studovaná křivka je dána podílem těchto dvou hodnot.

MSW se v průběhu snažíme minimalizovat a SSB maximalizovat a zastavit se u optimálního poměru. Tato technika by měla vést k lepšímu odhadu, nelze ji však v problematice adaptace použít. Důvod viz dále.

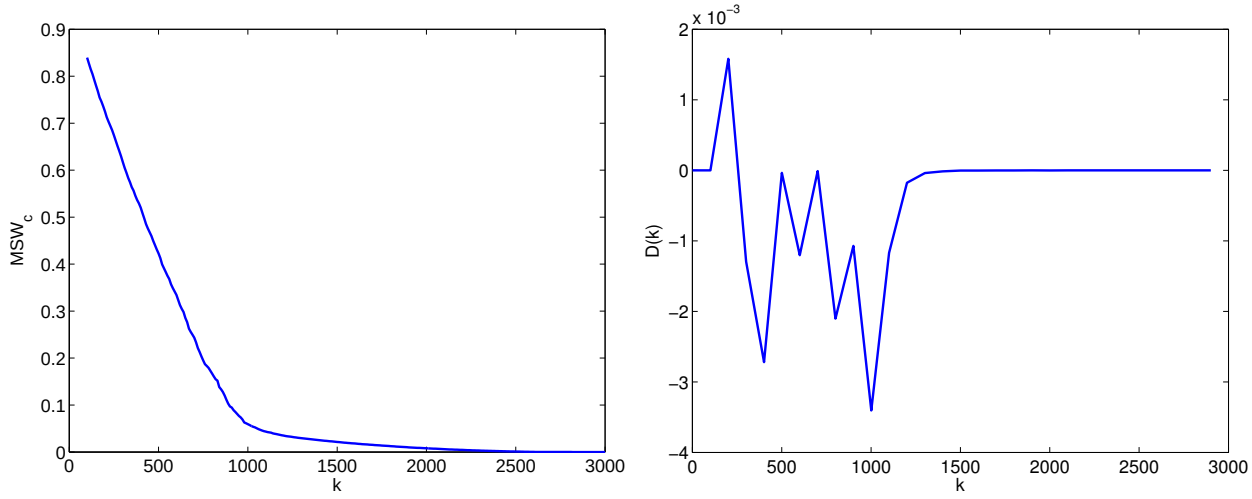
V běžném problému shlukování máme velké množství datových vektorů (stovky a tisíce) a pouze malé množství požadovaných shluků. V naší problematice je situace opačná ve výsledku hledáme velké množství shluků (mluvčích)a ke každému malé množství datových vektorů (jednotky, desítky). Časová složitost výpočtu SSB je poté neúnosná. V každé iteraci tak měříme vzdálenost všech kombinací stovek a tisíců shluků. V odhadu se proto omezuje pouze na MSW.

Změna směrnice přímky

Jedním z takových způsobů nalezení ohybu je například hledání nejmenší změny směrnice přímky ([9]), jenž je dáno vztahem 4.9. Odhadovaný počet shluků je pak dán hledáním minima v takovéto funkci. Testovaný průběh a jeho odezvu na zmíněnou funkci lze vidět na snímku 4.4.

$$D(k) = MSW(k)_{dif}^{d/2} - MSW(k-1)_{dif}^{d/2} \tag{4.9}$$

Tato technika ovšem vyžaduje, aby tato změna směrnice byla výraznější vůči svému okolí. To v případě MSE s Euklidovou vzdáleností není vždy patrné, jak je vidět na snímku 4.3.



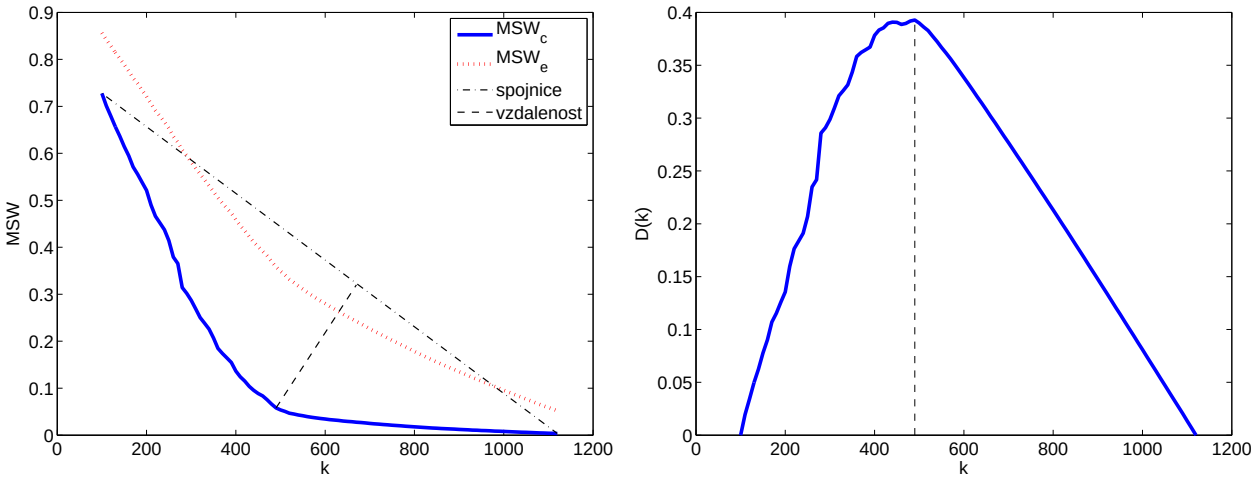
Obrázek 4.4: Ukázka vývoje MSW křivky a její diferenční funkce.

Negativem této metody je dále nutnost výraznější změny v rámci sousedních kroků. V případě testování v rámci několika málo desítek shluků, bude tento princip fungovat dle očekávání. V případě odhadování na velkém množství dat (tisíce a desetitisíce datových bodů), kde je zároveň velké množství shluků (tisíce), algoritmus drobně selhává (funkční křivka má příliš jemný krok). Ve hledaném ohybu není změna směrnice příliž velká v porovnání s blízkým okolím. Ve výsledku na místo dominantní minimální hodnoty z funkce 4.9

získáme řadu malých hodnot vzájemně blízkých, kde vlivem odchylky může být vybráno chybné místo určující optimální řešení. Tomuto problému se lze bránit snížením počtu bodů MSW křivky (například zvětšením kroku s počtem shluků).

Vzdálenost bodu přímky od spojnice koncových bodů

Pro nalezení bodu křivky odpovídajícímu odhadovanému optimálnímu řešení je zde hledán bod, jenž má od spojnice koncových bodů křivky největší vzdálenost ([19], obr. 4.5).



Obrázek 4.5: Ukázka vývoje funkce principu měření největší vzdálensoti od spojnice koncových bodů MSW křivky.

Na rozdíl od předchozí metody není tato metoda zatížená na relativně nízké změny směrnice vzhledem k blízkému okolí. Je také schopná dopracovat se použitelnějšího výsledku v případech, kdy celkový ohyb není tak výrazný (viz 4.3).

Úhlově založená metoda

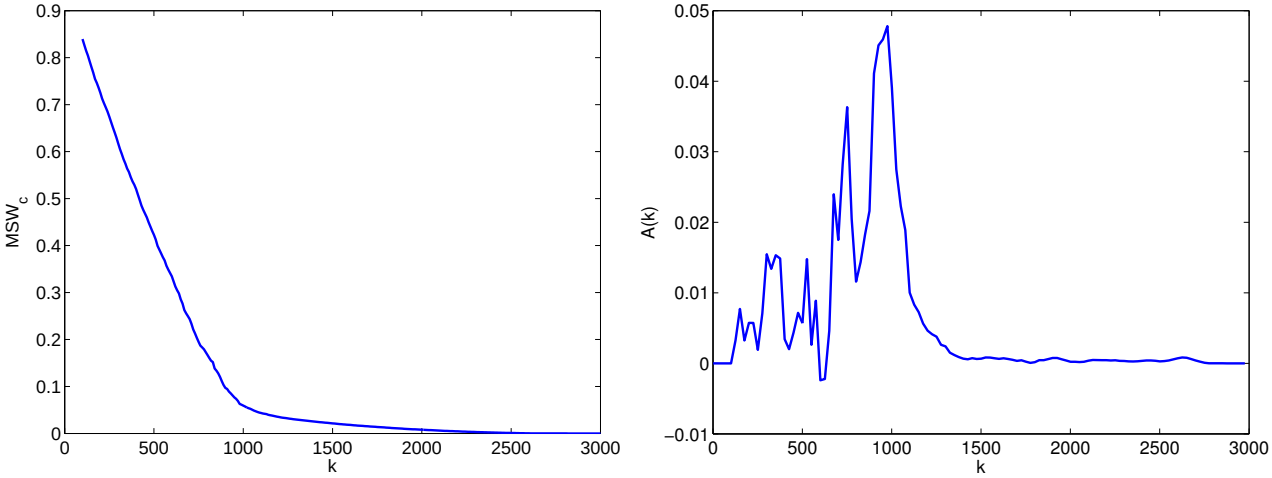
Metoda popsána v [25] je založena na prohledání lokálních změn křivosti v obou směrech (ne pouze minulé a současné hodnoty). Díky tomu je méně závislá na drobné odchylky v rámci jednoho kroku. Díky pevně danému aktuálně zkoumanému bodu, lze pomocí změny konstanty c v předpisu funkce 4.10 také regulovat náchylnost na drobné odchylky tím, že roztáhneme ramena zkoumaného okolí.

$$A(k) = MSW(k-c) + MSW(k+c) - 2MSW(k) \tag{4.10}$$

Optimální řešení je dáno maximální hodnotou funkce (viz obr 4.6).

4.5.2 Odhad počtu shluků pro k-means

Tento algoritmus publikovaný v [15] je podobou příbuzný s odhadem počtu shluků pomocí změny směrnice. Používáme ovšem odlišnější funkci (ovšem obdobného průběhu) a ohyb je detekován jiným způsobem. K získání odhadu se využívá vývoje křivky $S_k = \sum_{j=1}^{K} I_j$,



Obrázek 4.6: Ukázka vývoje MSW křivky a její odezvy na 4.10.

dané celkovou vzdáleností ve shlucích $I_j = \sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\mu}_{pi}\|$, kde $\mathbf{x}_i$ odpovídá jednotlivým datovým vektorům a $\boldsymbol{\mu}_{pi}$ střední hodnotě odpovídajícího shluku. V tomto vývoji se pomocí funkce 4.11 hledá patřičná změna. Zvolený odhad poté odpovídá hodnotě, které přísluší minimum v dané funkci.

$$f(k) = \begin{cases} 1 & k = 1 \\ 1 & S_k = 0, \forall > 1 \\ \frac{S_k}{\alpha_k S_{k-1}} & S_k \neq 0, \forall > 1 \end{cases} \tag{4.11}$$

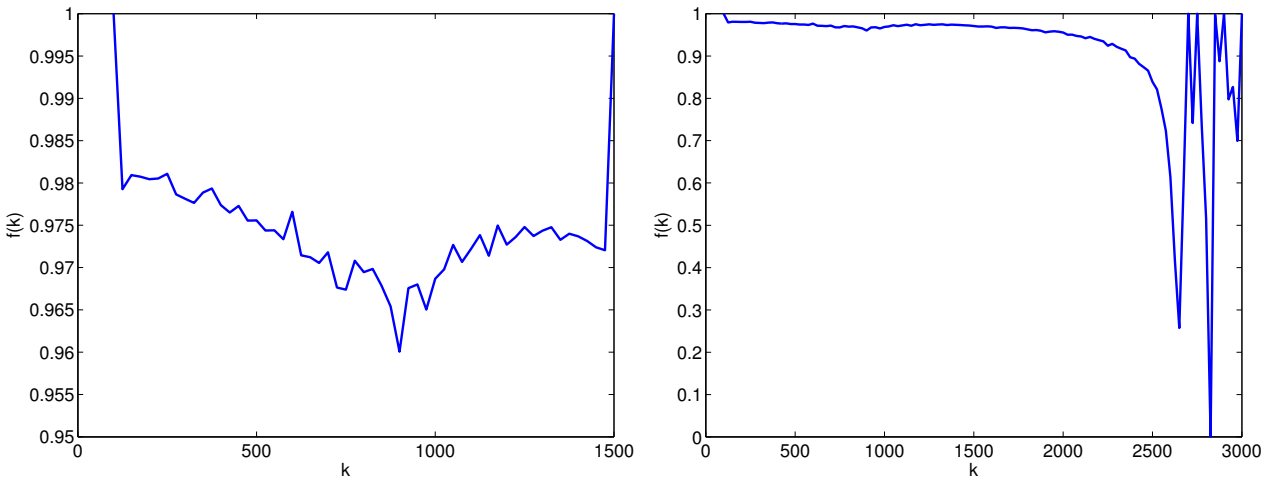
$$\alpha_k = \begin{cases} 1 - \frac{3}{4N_d} & k = 2, N_d > 1 \\ \alpha_{k-1} + \frac{1-\alpha_{k-1}}{6} & k > 2, N_d > 1 \end{cases} \tag{4.12}$$

Koeficient α_k slouží jako korekční koeficient. Pomáhá udržet úroveň funkce $f(k)$ při lineárně klesající křivce S_k. V případě jeho absence by se vlivem snižování hodnot v podílu nechtěně snižovala celková křivka $f(k)$. Nevýhodou algoritmu je jeho nestabilita v oblastech, kde se počet shluků blíží počtu dat. V těchto oblastech dosahují hodnoty S_k již malých hodnot (většina shluků je tvořena jediným prvkem). Postupným testováním většího počtu shluků je více shluků tvořeno jediným prvkem. Mají tedy nulovou hodnotu funkce I_k a to vede na řádové změny mezi jednotlivými hodnotami S_k, což se negativně projeví ve výsledné funkci $f(k)$.

Na snímku 4.7 je vidět požadovaný průběh, v porovnání s průběhem při výše popsané situaci. Z uvedeného vyplývá, že už při volení rozsahu v jakém se má množství shluků hledat, musí být nějakým způsobem omezeno. Na rozdíl od dříve popsaných metod, kdy jsme mohli testovat až do celkového množství dat.

4.6 Vyhodnocení metod na odhad počtu shluků

Metody byly testovány na dvou počtech shluků, opět je jím 500 a 1000 mluvčích. V jednotlivých iteracích se postupně navyšoval počet vektorů připadajících na jednoho mluvčího,



Obrázek 4.7: Ukázka vývoje funkce $f(k)$ V případě omezeného počtu testů (vlevo) a neomezeného počtu testů (vpravo).

v rozmezí $2, 4, 6, 8, 10$. Přírůstek použitý pro konstrukci MSW křivky byl 25 shluků. V každé iteraci bylo provedeno několik pokusů, z nichž byla vypočítána průměrná chyba odhadu. Dosažené výsledky lze vidět na obrázku 4.8.

Celkové výsledky průměrných chyb lze vidět v tabulce 4.3. V chybě některých metod se jistě projevuje fakt, že předpokládají podstatné vyšší množství vzorků dat připadajících na jeden shluk. V našem případě si v problematice adaptace a rozpoznávání mluvčích můžeme dovolit předpokládat jednotky, maximálně desítky, nahrávek (vektorů) na jednoho mluvčího. To je v rozporu s obvyklým problémem řešeným pomocí shlukování, kde se pracuje se stovkami a tisíci datovými vektory na shluk (týká se především metody 4.5.2).

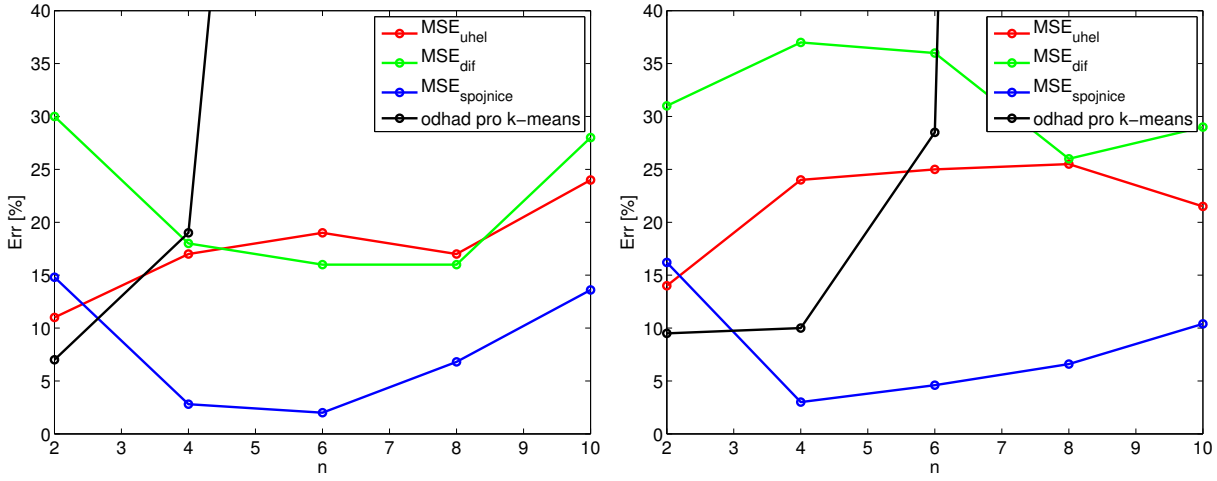
skutečný počet shluků	MSW_{dif}	$MSW_{spojnice}$	MSW_{uhel}	odhad pro k-means
500	21.6%	8.0%	17.6%	257.4%
1000	31.8%	8.2%	22.0%	232.4%

Tabulka 4.3: Srovnání metod odhadu počtu shluků pomocí celkové průměrné chyby odhadu.

Metoda odhadu počtu shluků pro k-means si vlivem dříve zmíněného problému vedla naprosto nejhůře, při testech nebyl dostatečně omezen rozsah v jakém měla metoda testovat. Z počátku ovšem dosahovala, jak je vidět na snímku 4.8, nejlepších výsledků. Své místo by proto mohla nalézt v situacích, kdy je nám alespoň přibližně znám počet (rozsah) shluků a chceme jen nalézt přesnější hodnotu.

Stejně jako u shlukovacích algoritmů byly tyto techniky na odhad počtu shluků (mluvčích) v datové sadě testovány na ideálním stavu, tedy na základě podobnosti systému trénovaného v doméně. Víme tedy, čeho jsou metody schopny dosáhnout. Pro srovnání opět uvedeme stav, jenž odpovídá počátečnímu kroku adaptace, kdy je podobnost získána z neadaptovaného systému (tabulka 4.4).

V případě užití mimo-doménového skóre jako virtuální podobnosti se celkové chyby metod odhadu počtu shluků předpokládaně zvýšily. Metoda založená na zkoumání vzdálenosti



Obrázek 4.8: Chyba odhadů počtu shluků pro 500 mluvčích (vlevo) a 1000 mluvčích (vpravo).

skutečný počet shluků	MSW_{dif}	$MSW_{spojnice}$	MSW_{uhel}	odhad pro k-means
500	22.4%	33.4%	28.8%	272.0%
1000	21.8%	37.0%	20.0%	230.7%

Tabulka 4.4: Výsledky odhadu počtu shluků pomocí celkové průměrné chyby odhadu (při odhadu z podobnosti získané z mimo-doménového systému).

bodů od spojnice koncových bodů MSW křivky si již nevedla nejlépe, přesto bude nejvhodnější ji dále použít. Tato metoda je stabilnější a nezávislá na počtu bodů křivky. Oproti ostatním se její chybovost tolik nemění v závislosti na jemnosti výpočtu křivky (přírůstku počtu shluků použitý při konstrukci MSW křivky). Důležité také je, že zmíněná chyba u metody s výsledkem v $MSW_{spojnice}$ se téměř vždy projevila chybným určením většího množství shluků než bylo jejich skutečné množství (oproti ostatním metodám, které odhadovaly nižší počty). Toto nám umožňuje s následně odhadnutými identitami mluvčích dále pracovat (viz filtrace v 4.7).

Odhad z virtuální vzdálenosti

Jak bylo zmíněno dříve, u odhadu provádíme shlukování pomocí virtuální podobnosti získané z PLDA systému a pro odhad počtu mluvčích užíváme jiné metriky (MSW) . Důvodem byla možnost měřit vzdálenost bodu od středu příslušného shluku. Ukažme si nyní, proč bylo výhodnější použít této kombinace místo samotného užití virtuální podobnosti.

Při užití samotné podobnosti víme pouze relativní vzdálenost mezi jednotlivými kombinacemi vektorů, toho nelze využít u měření vzdálenosti vůči určitému středu. Při měření průměrné vzdálenosti uvnitř shluku se tedy musíme omezit na průměrnou vzdálenost mezi jednotlivými vektory,místo mezi vektorem a středem. Na takto vzniklou křivku aplikujeme algoritmus odhadu podle největší vzdálenosti od spojnice koncových bodů (uvedené v části 4.5.1), která se jeví jako nejvhodnější. Virtuální podobnost byla získána ze systému tréno-

vaného na doménových i mimo-doménových datech.

skutečný počet shluků	$MSW_{spojnice\_virt}$
500	13.0%
1000	13.1%

Tabulka 4.5: Výsledky odhadu počtu mluvčích s metodou nejdelší vzdáleností od spojnice koncových bodů pouze s pomocí virtuální podobností (vzdáleností) v patřičné doméně.

skutečný počet shluků	$MSW_{spojnice\_virt}$
500	38.0%
1000	40.9%

Tabulka 4.6: Výsledky odhadu počtu mluvčích s metodou nejdelší vzdáleností od spojnice koncových bodů pouze s pomocí virtuální podobností (vzdáleností) získané z mimo-doménového systému.

4.6.1 Kvalita shlukovacích algoritmů ve spojení s odhadem počtu shluků

Jelikož v práci jde o kombinaci odhadu počtu mluvčích s odhadem jejich počtu přidejme i výsledky kvality shlukování, kdy nebyl znám přesný počet shluků a pro jeho zjištění byl využit odhad metodou založenou na vzdálenosti od koncových bodů MSW křivky.

Výsledky vidíme v tabulce 4.7. Klíčovým údajem je zde především f-skóre, protože během experimentů přesnosti odhadu pomocí mimo-doménového systému bylo zjištěno, že použitá metoda má tendenci vždy určit větší množství shluků než je skutečné.

	Mluvčích	**Čistota shluku**	**f-skóre**
Spojování komponent	500	0.92	0.86
	1000	0.91	0.84
Aglomerativní h. shlukování	500	0.90	0.83
	1000	0.88	0.80

Tabulka 4.7: Srovnání shlukovacích algoritmů v souvislosti s provedeným odhadem počtu shluků (jako podobnost bylo použito skóre z mimo-doménového systému).

Oproti výsledkům z tabulky 4.2 se může zdát, že jsou výsledky lepší (přesto, že v uvedené tabulce odpovídají situaci se znalostí správného počtu shluků), je to pouze zdání. Uvedená metoda odhaduje s užitím mimo-doménových dat vždy větší počet shluků. Což se projevuje virtuálním zlepšením celkového odhadu. Podrobněji tuto problematiku ještě zhodnotíme později, kdy budeme měřit vliv kvality odhadu na chybu systému (6.3).

Na základě těchto údajů vyplynula potřeba nalézt způsob zvýšení kvality odhadu ve spojení s odhadem počtu.

4.7 Zvýšení kvality odhadu identit mluvčích

Zvýšení kvality odhadu identit je založeno na poměrně jednoduchém principu. Ze sady dat u které jsme prováděli odhad mluvčích, jednoduše vybereme ty mluvčí, kteří byli odhadnuti správně. Přesněji řečeno ty, u nichž je pravděpodobnější, že byli odhadnuti správně. Zaměřme se nyní tedy na to, jak tyto mluvčí vybrat a podle čeho je poznat.

Při odhadu identit mluvčích v adaptaci využíváme na sobě závislých technik odhadu počtu shluků (mluvčích) a následného shlukování (odhad identit mluvčích). Obě techniky jsou zatíženy jistou chybou, která ve výsledku po jejich spojení jistě značně vzroste (akumuluje se).

Z povahy problému odhadu identit víme, že existuje jisté přesné a správné řešení, kdo jakou nahrávku reprezentovanou i-vektorem namluvil. Dále také víme, že existují logické a fyzické limity vstupních dat. Představme si pod tím situaci, kdy po provedení odhadu je velké množství dat (stovky, či tisíce nahrávek) přiřazeno jednomu mluvčímu. Fyzicky jistě není nemožné namluvit takovéto množství nahrávek, je ale velmi nepravděpodobné, že by je shlukovací algoritmus odhadl takto dobře a že by se takové množství dat dostalo do datové sady. Tyto velké shluky můžou vznikat především v případech, kdy je chybně odhadnut menší počet mluvčích oproti skutečnému počtu.

Příliš velké shluky budou tedy pravděpodobně chybně zařazené nahrávky, z trénovací sady je proto odstraníme. Zbavíme se tak sice části trénovacích dat, zároveň, ale odstraňujeme data, která by vedla k chybnému natrénování.

Nyní se podívejme na odhadnuté identity z druhé strany a to jsou příliš malé shluky. Především v případech, kdy odhadneme větší množství mluvčích, než je skutečné, se tato chyba projeví právě vznikem menších shluků (normálně velký shluk je rozdělen na několik menších). S jistotou nelze říci, zda malý shluk je vzniklý chybně odhadnutým počtem nebo opravdu je od jistého mluvčího pouze několik nahrávek v datové sadě. Při znalosti PLDA víme, že pro její dobré natrénování potřebujeme dostatečné množství nahrávek od každého mluvčího. Mluvčí, kteří mají po odhadu identit jen velmi málo nahrávek (např. pouze jedinou), nepřinášejí do trénování dostatečně užitečnou informaci, proto je z trénovací sady také odstraníme.

Zvýšení kvality odhadu identit tedy spočívá ve filtraci získaných výsledků, kdy pro další zpracování vybereme pouze optimálně velké a reálné shluky dat. Snížíme sice velikost trénovací sady pro další použití, zároveň tím odstraňujeme pravděpodobně chybně klasifikovaná data, která by zanášela do trénování další chybu.

4.7.1 Vliv filtrování na kvalitu odhadu identit

Pro lepší představu si ukažme vliv filtrace na výslednou kvalitu odhadu identit mluvčích. Pro přehlednost rozlišme tyto pojmy:

- skuteční mluvčí – Takto označíme mluvčí, kteří jsou v datové sadě podle anotací o jejich identitách.
- virtuální mluvčí – Takto označíme mluvčí, které jsme odhadli pomocí shlukovacích algoritmů.

Pro úplnost si připomeňme, že bude pravděpodobně rozdíl v identitách skutečných a virtuálních mluvčích a i v jejich počtu.

Výsledky filtrace si ukážeme na datové sadě 10000 i-vektorů od 1000 mluvčích (v naší terminologii skutečných mluvčích). Vektory jsou od mluvčích vybrány rovnoměrně, tedy od každého mluvčího máme právě 10 i-vektorů.

Pro samotnou filtraci jsme vybrali dva rozsahy. V prvním budeme z datové sady vybírat pouze ty virtuální mluvčí, kteří mají přiřazeno 10 i-vektorů, tedy stejně jako skuteční mluvčí. Tento test nám neprozradí příliš o vhodnosti filtrace pro následné trénování PLDA. Díky němu, si ale můžeme udělat představu o skutečné kvalitě samotného shlukování v závislosti na odhadu počtu identit. Získáme tak lepší představu než z chyby odhadu počtu a identity zvlášť. Výsledky nalezneme v tabulce 4.8.

	před filtrací	po filtraci
počet skutečných mluvčích	1000	429
počet virtuálních mluvčích	1660	416
f-skóre odhadu	0.827	0.994
čistota odhadu	0.949	0.996
celkový počet i-vektorů v sadě	10000	4163

Tabulka 4.8: Výsledky filtrace při výběru pouze virtuálních mluvčích se stejným počtem vektorů jako měli skuteční mluvčí (10).

Z výsledku této filtrace lze vyčíst několik podstatných informací. V první řadě vidíme, že pouze necelé polovině virtuálních mluvčích bylo přiřazeno přesně tolik vektorů, kolik měli skuteční mluvčí. Filtrací jsme tak přišli o velké množství dat. Zároveň si povšimněme, že počet virtuálních mluvčích je po filtraci velmi blízký počtu skutečných mluvčích, kteří v datové sadě zůstali. Filtrace tedy poměrně dobře kompenzovala chybný odhad mluvčích v sadě. Celkově, jak je vidět na čistotě odhadu a f-skóre, kvalita odhadu značně stoupla.

Nyní se podívejme na stejný odhad, nyní s filtrací v rozsahu $\langle 5, 15 \rangle$, tedy tak, že vybraní mluvčí měli minimálně 5 a maximálně 15 přiřazených vektorů (tabulka 4.9).

	před filtrací	po filtraci
počet skutečných mluvčích	1000	938
počet virtuálních mluvčích	1660	935
f-skóre odhadu	0.827	0.970
čistota odhadu	0.949	0.964
celkový počet i-vektorů v sadě	10000	8129

Tabulka 4.9: Výsledky filtrace při výběru virtuálních mluvčích s minimálním počtem vektorů 5 a maximálním 15.

S mírnějším výběrem mluvčích jsme nedosáhli takového zlepšení oproti předchozímu případu. Zároveň jsme ovšem dosáhli velmi blízkého přiblížení počtu virtuálních mluvčích k počtu skutečných mluvčích. Dále také pokles počtu i-vektorů v sadě je o dost nižší. Zůstává nám tedy více dat k dalšímu zpracování.

Pro představu si ještě uveďme situaci, kdy jsme shlukovacímu algoritmu dali informaci o přesném počtu mluvčích. Zkoumáme tedy samotnou metodu odhadu identit (tabulka 4.10).

	před filtrací	po filtraci
počet skutečných mluvčích	1000	851
počet virtuálních mluvčích	1000	732
f-skóre odhadu	0.770	0.908
čistota odhadu	0.762	0.892
celkový počet i-vektorů v sadě	10000	7032

Tabulka 4.10: Výsledky filtrace při výběru virtuálních mluvčích s minimálním počtem vektorů 5 a maximálním 15 v případě shlukování do správného počtu shluků.

Opět došlo ke zlepšení výsledného odhadu. Filtrace má tedy své místo nejen v kombinaci technik odhadu počtu a identit, ale i v odhadu identit samotném. Matoucí se může jevit hodnota čistoty shluku a f-skóre v případě, kdy bylo shlukováno do správného počtu shluků. Kvalita shlukování se jeví jako nižší než v situaci, kdy jsme provedli chybnější odhad. Nižší přesnost je pouze zdánlivá, vzpomeňme na průběh shlukovacích evaluačních metrik z podkapitoly 4.2, kdy výsledná kvalita klesá podstatně pomaleji při vyšším množství shluků, než při nižším (oproti skutečnému).

Později se podíváme i na vliv takovéto filtrace na celkovou chybu systému, kdy se filtrovaná sada samotná použila k natrénování PLDA. V příloze B.2 lze nalézt kompletní testy filtrace pro testovací sady s 500, 1000 a 3799 (celá dostupná datová sada) mluvčích. Testy obsahují vliv filtrace na kompletní odhad identit (odhad počtu a shlukování) i samotného shlukování pro dané datové sady.

4.7.2 Dvojitá filtrace

Adaptační technika uvedená v části 3.3.2 popisuje rozdílné nároky při váhování modelu ve variabilitě mluvčích a kanálu. Je zde popsána možnost různého váhovaného poměru. Zkusili jsme tuto skutečnost zohlednit při filtrování.

Při trénování modelu PLDA tedy opět provádíme filtraci výsledků odhadu identit mluvčích. Části modelu (pro variabilitu mluvčích a kanálu) jsou trénovány odděleně na různě filtrovaných odhadech mluvčích.

Variabilita mluvčích je filtrovaná přísněji. Naproti tomu je filtrace pro trénování variability kanálu nepatrně mírnější. Interval filtrace je o několik jednotek rozšířen po obou stranách. To ve výsledku vede k možnosti širší kovarianci kanálu.

Trénovací sada pro trénování AC variability byla opět filtrována na intervalu $\langle 5, 15\rangle$ výsledky v tabulce 4.9). Pro trénování WC variability byl interval, jak již bylo zmíněno, rozšířen na $\langle 3, 20\rangle$. Výsledná filtrovaná trénovací sada pro trénování WC variability dosahovala výsledku v tabulce 4.11.

	před filtrací	po filtraci
počet skutečných mluvčích	1000	991
počet virtuálních mluvčích	1660	1157
f-skóre odhadu	0.827	0.902
čistota odhadu	0.949	0.954
celkový počet i-vektorů v sadě	10000	9182

Tabulka 4.11: Výsledky filtrace při výběru virtuálních mluvčích s minimálním počtem vektorů 3 a maximálním 20 užitého u dvojí filtrace.

Kapitola 5

Evaluační datové sady a základní skóre

5.1 Datové sady

Pro trénování a evaluace rozpoznávacího systému je použito několik datových korpusů. Jako zdroj dat pro mimo-doménový systém slouží korpus **Switchboard** (Fáze 1, Fáze 2 a Cellular). Jako adaptační data jsou použité nahrávky z korpusů **NIST** z let 2004, 2005, 2006 a 2008 ([2]). Jako evaluační sada poté slouží korpus **NIST** 2010.

K dispozici je ve výsledku 32921 nahrávek (1461 mužských a 1653 ženských) z korpusu **Switchboard**. Pořízených se vzorkovací frekvencí $8000Hz$ a vzorkovacím formátem *dvoukanálový μ-law*. Z korpusu **NIST** pak obdobně disponujeme 36498 nahrávkami (1531 mužských a 2276 ženských).

5.2 Základní skóre

Základní skóre sloužící pro vyhodnocení vytvořených metod. Jako tato skóre byly vybrány výsledky dvou hraničních případů. Prvním případem jsou výsledky mimo-doménovým neadaptovaným systémem, jenž určuje spodní hranici výsledků. Druhým případem jsou výsledky systému adaptovaného pomocí adaptace s učitelem, tedy případu, kdy jsme měli identifikaci jednotlivých mluvčích v adaptačních datech. Zvolené výsledky nám vymezují interval, ve kterém se budeme pohybovat s implementovanými metodami, zároveň se nepředpokládá, že bychom jeho hranice překročili.

Získané výsledky pro dolní hranici lze nalézt v tabulce 5.1. Pro horní hranici v tabulce 5.2.

gender	**newmindcf**	**oldmindcf**	**EER**	**targets**	**nontargets**
	0.73	0.30	6.6%	7169	408950
f	0.76	0.32	7.51%	3704	233077
m	0.68	0.28	5.58%	3465	175873

Tabulka 5.1: Dolní hranice základního skóre.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.47	0.13	2.57%	7169	408950
f	0.50	0.14	2.76%	3704	233077
m	0.42	0.11	2.29%	3465	175873

Tabulka 5.2: Horní hranice základního skóre.

Obě tabulky odpovídají vyhodnocovacímu výstupu systému. Význam jednotlivých prvků je následující:

- **f**,**m**: označují pohlaví mluvčích. Ženy jsou označeny **f** (female), muži **m** (male), prázdné políčko označuje obě pohlaví.
- **newmindcf**,**oldmindcf**: označují evaluační metriky DFC (Detection Cost Function), představené NIST (Nation Institute of Standart and Technology). Popsány v [1].
- **EER**: označuje chyby systému Equal Error Rate (viz 5.2.1).
- **targets**, **nontargets**: označují počet pozitivních **targets** a negativních **nontargets** testů.

5.2.1 EER – Equal Error Rate

Při měření vlivů různých parametrů a technik jsme se soustředili na změny hodnot EER, neboli Equal Error Rate. Tato chyba udává chybu systému v bodě, kdy pravděpodobnost falešné detekce (v angličtině nazývaná *false alarm* – P_{FA}) je rovna pravděpodobnosti opomenuté detekce (v angličtině nazývaná *miss* – P_{miss}). Pravděpodobnost falešné detekce udává pravděpodobnost, že dvojice nahrávek od různých mluvčích (nontargets) je označena jako od stejného mlučího. Pravděpodobnost opomenuté detekce je pravděpodobnost, že dvojice nahrávek od stejného mluvčího (targets) je označena jako od různých mluvčích (více viz [20]).

Kapitola 6

Sestavení adaptačního systému

V předchozích kapitolách jsme se zaobírali teoretickými možnostmi adaptace a dalšími technikami při adaptaci použitých (shlukování, odhad počtu mluvčích). V případných měřeních jsme využívali metriky z dané problematiky (například čistota shluků). V této kapitole se zaměříme na zjištění vlivu chybovosti jednotlivých kroků na celkovou chybovost systému (EER). Na základě těchto zjištění provedeme sestavení jednotlivých kroků adaptace a provedeme porovnání mezi jednotlivými adaptačními technikami.

6.1 Ukončovací kritérium

U většiny zmíněných adaptačních technik provádíme iterace za účelem zvyšující se přesnosti odhadu identit a tím postupného klesání výsledné chyby systému. V takovémto případě potřebujeme poznat, kdy je vhodné adaptaci ukončit.

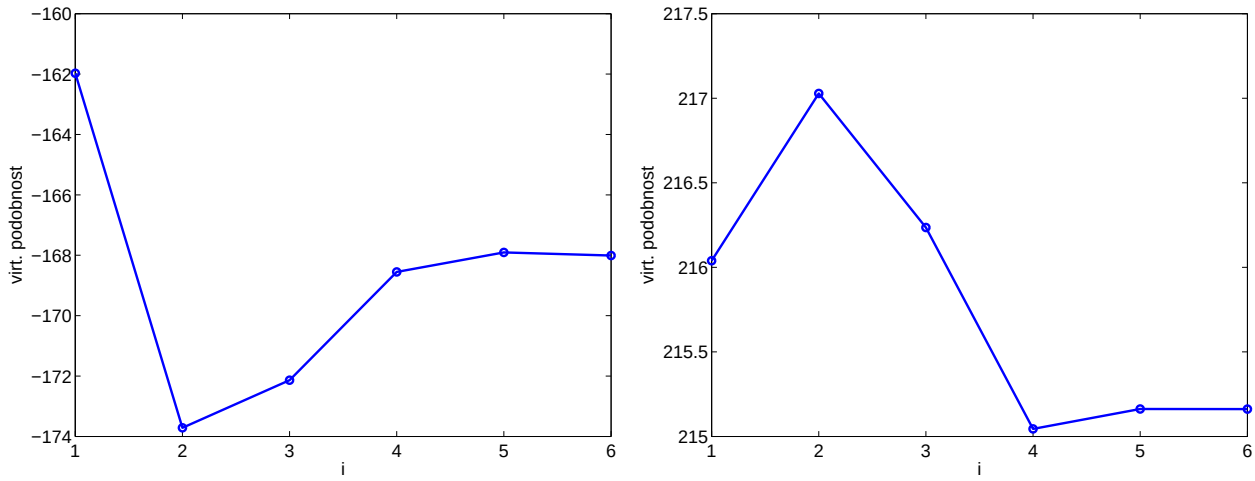
Ukončení adaptace by mělo proběhnout v okamžiku, kdy jsme dosáhli nejmenší výsledné chyby systému a již je jisté, že nebude tato chyba dále klesat.

V okamžiku adaptace (a ani po něm) nemáme k dispozici informaci o tom, jaká je výsledná chyba systému a jak se nám mění. V podstatě jediné, co vždy máme mezi jednotlivými iteracemi, jsou adaptační i-vektory a skóre PLDA systému s jejich podobností.

První možností, která nás může v souvislosti s ukončením iterativního výpočtu napadnout, je konvergence nějaké metriky k určité hodnotě. V tomto ohledu by se tak mohla jevit podobnost adaptačních i-vektorů. Bohužel tomu tak není. Vysvětlení je prosté. Výstupní skóre PLDA systému se sice v každé iteraci mění a postupně zpřesňují informaci o podobnosti i-vektorů, ale nekonvergují k žádné hodnotě.

Skóre systému je závislé na modelu, který je k jeho vyprodukování použit. My ovšem v každé iteraci tento model měníme. Důsledkem toho se kompletně mění poměr mezi jednotlivými hodnotami skóre, celkové maximum, minimum i rozsah obsažených hodnot. Účelem adaptace je lepší rozpoznání i-vektorů, a tedy i větší rozdíly v hodnotách skóre mezi jednotlivými i-vektory. Z těchto důvodů skóre neposkytuje tedy přímo informaci o ukončení iterací. Vývoj podobnosti lze vidět na obrázcích 6.1.

Hodnoty skóre nám tedy svým průběhem neposkytují vodítko k tomu, abychom poznali, že je čas adaptaci ukončit. Při uvážení adaptačních technik si můžeme všimnout na čem jsou všechny iterativní přístupy závislé. Závislost spočívá v odhadu identit mluvčích v datové sadě. Mezi jednotlivými iteracemi se tento odhad identit bude nutně měnit. Z tohoto vyplývá, že v okamžiku, kdy se tento odhad měnit přestane, dosáhli jsme momentu, kdy se systém adaptací přestává zlepšovat a je vhodné adaptaci ukončit. Bez měnících se iden-



Obrázek 6.1: Vývoj hodnot virtuální podobnosti (PLDA skóre) dvou rozdílných vektorů (vpravo) a vektoru se sebou samým (vlevo).

tit mluvčích v adaptační sadě totiž již nedojde k podstatné změně modelu a tím i celého systému.

Pro zjištění, jestli a jak moc se identity mění, využijeme již zmíněné metriky pro měření kvality shlukování. Zmínili jsme dvě tyto techniky. První je *Čistota shluku* a druhou *f-skóre*. Pro odhalení, zda se identity ještě mění, můžeme použít pouze f-skóre.

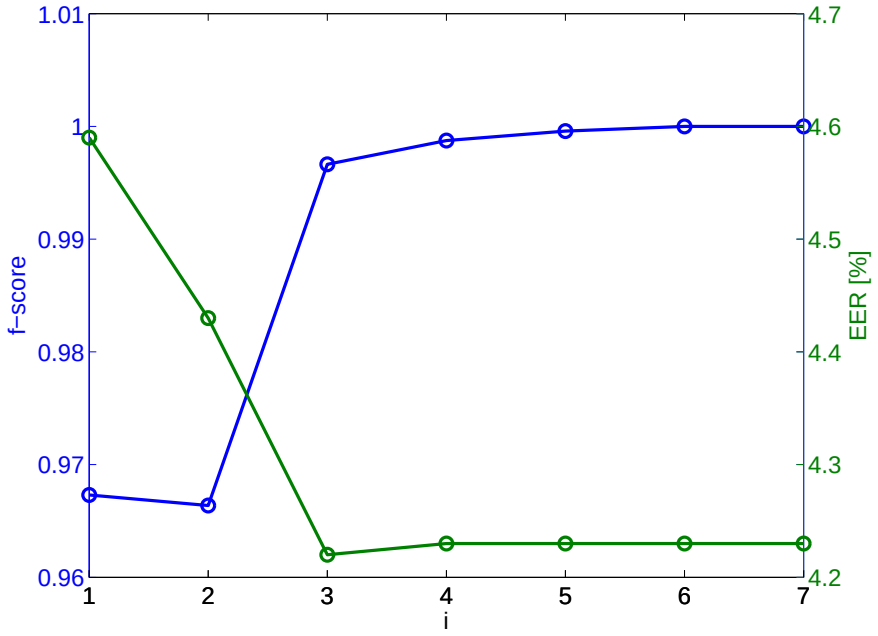
Čistotu shluku nelze použít kvůli její neefektivnosti v okamžiku, kdy je odhadnutý počet shluků vyšší než skutečný. Odhad počtu shluků probíhá také v každé iteraci a i počet shluků se postupně zpřesňuje. Počet shluků se tedy mezi iteracemi může měnit. Je proto nutné použít metriku, která rozdílný počet zohlední a touto metrikou je právě f-skóre. Na jeho hodnotách je dobře patrný rozdíl odhadu identit mezi jednotlivými iteracemi, a to i za situace, kdy se odhadnutý počet mluvčích mezi iteracemi liší. Průběh hodnoty f-skóre lze vidět na obrázku 6.2 společně s vývojem celkové chyby systému. Pro zobrazené průběhy je využito adaptačního průběhu metody *Shlukování a přetrénování*.

Ze snímku 6.1 jasně vidíme, že ať se skóre nakonec přeci jen v posledních dvou hodnotách ustálilo (přesto to neznamená, že by se pro ukončovací kritérium dalo použít), samotná adaptace podle chyby systému skončila daleko dříve (viz 6.2).

6.2 Výběr shlukovacího algoritmu

Na základě porovnání metrik určených pro vyhodnocení úspěšnosti shlukování v části 4.2 již víme, že metoda Spojování komponent se jeví jako úspěšnější. Provedené měření ovšem odpovídá obecnému použití shlukovacích algoritmů. Provedli jsme proto měření, kdy jsme sledovali výslednou chybu systému. Ověříme, tak zda úspěšnost odhadu v obecném pojetí bude odpovídat i v problematice adaptace a zda opravdu úspěšnější algoritmus bude dávat lepší výsledky systému.

Test probíhal přímým trénováním PLDA systému na adaptačních datech. Oba shlukovací algoritmy vždy prováděly shlukování na stejné datové sadě a tedy i se stejným odhadem počtu mluvčích. V této části nebyly výsledné odhady filtrovány, neboť jsme chtěli sledovat



Obrázek 6.2: Vývoj hodnoty f-skóre a odpovídající hodnoty EER systému.

právě samotné shlukovací algoritmy.

Test je následující:

1. Odhad počtu mluvčích v adaptační sadě pomocí skóre mimo-doménového systému.
2. Odhad identit mluvčích pomocí skóre mimo-doménového systému a počtu z kroku 1.
3. Trénování nového PLDA modelu.
4. Vyhodnocení evaluační sady.

Principielně tento test odpovídá prvnímu kroku adaptace, kdy k trénování modelu použijeme pouze adaptační i-vektory.

algoritmus	**EER**
Spojování komponent	5.18%
Aglomerativní hierarchické shlukování	5.48%

Tabulka 6.1: Výsledky systému v závislosti na shlukovacím algoritmu.

Z uvedených výsledků (tabulka 6.1) jsme zjistili, že Spojování komponent je opravdu v odhadech identit úspěšnější. Podle rozdílů výsledků Spojování komponent a Aglomerativního hierarchického shlukování lze usuzovat, že hodnoty f-skóre budou mít přímo úměrný vliv na výslednou chybu systému.

6.3 Vliv filtrace na chybu systému

Podívejme se nyní, jestli zvýšení kvality odhadu identit opravdu povede ke snížení chyby systému. Tyto testy nám prozradí, jestli zvýšení kvality odhadu identit mluvčích na úkor velikosti trénovací sady opravdu povede ke snížení chyby systému, či zda se snížení počtu trénovacích vektorů projeví příliš negativně. Průběh testu je shodný jako při výběru shlukovacího algoritmu s tím rozdílem, že odhadnuté identity jsou filtrovány.

V této části byly prověřeny dva filtrované rozsahy. Systém s přísnou filtrací, kde jsme opět vybírali pouze virtuální mluvčí, jimž byl přiřazen stejný počet vektorů, jako měli skuteční mluvčí. Druhým systémem byl systém pro jehož trénování byli vybráni virtuální mluvčí, kterým byl přiřazen počet vektorů v intervalu $\langle 5, 15\rangle$. Tento rozsah byl zvolen, neboť se jedná o 50% toleranci ke skutečnému počtu (který ovšem v reálných podmínkách nevíme), zároveň se ale jedná o možné a reálné počty, které by se mohly v trénovací sadě vyskytovat. Průběh testu je opět stejný jak již bylo uvedeno výše. Adaptační sada, použitá pro trénovaní, byla tvořena tisícem skutečných mluvčích po 10 i-vektorech, stejně jak tomu bylo u testech filtrace v 4.7.1.

rozsah filtrace	**EER**
bez filtrace	6.00%
$\langle 10, 10\rangle$	6.30%
$\langle 5, 15\rangle$	5.18%

Tabulka 6.2: Výsledky systému v závislosti na filtraci trénovací sady. Uvedené výsledky jsou pro kombinovanou evaluační sadu (ženy a muži dohromady).

Na výsledcích v tabulce 6.2 vidíme, že bez filtrace se systém trénovaný pouze na adaptační sadě zlepšil jen nepatrně. Projevuje se zde patrně právě kombinace chyb odhadu počtu i identit mluvčích. U přísné filtrace je výsledek horší než bez filtrace. Vzniklá chyba bude stále na vině lehce chybného odhadu v identitách, které nám pro trénovaní zůstaly a dále přílišným omezením datové sady (jak je vidět v tabulce 4.8).

U filtrace jsme dále zmínili možnost trénovat variability PLDA systému zvlášť na různě filtrovaných trénovacích sadách. Uveďme tedy i dosažené výsledky této možnosti. Pro trénování variability kanálu jsme o několik možných vektorů rozšířili požadovaný interval na $\langle 3, 20\rangle$ (výsledky v tabulce 6.3).

variabilita	**rozsah filtrace**	**EER**
AC	$\langle 5, 15\rangle$	5.10%
WC	$\langle 3, 20\rangle$	

Tabulka 6.3: Výsledky systému v závislosti na rozdílné filtraci trénovací sady pro AC a WC variabilitu. Uvedené výsledky jsou pro kombinovanou testovací sadu (ženy a muži dohromady).

6.4 Zhodnocení adaptačních technik

6.4.1 Shlukování a přetrénování

Nejjednodušší technika zmíněná v části 3.3.1. V této technice není nutné provádět žádné manipulace s modelem. Na základě zmíněného algoritmu 1 máme několik možností v kombinaci trénovací sady. Zmíněny jsou dva základní principy, konstantní podíl mimo-doménových dat, nebo snižující se.

Na tomto základě byly vytvořeny čtyři varianty systému:

1. $\eta = 0$ – adaptovaný systém bude trénován pouze na adaptační sadě
2. $\eta = 0.5$ – částečná kombinace variabilit
3. $\eta = 1$ – plný mimo-doménový systém rozšiřován adaptační sadou
4. $\eta = 1$ a v každé iteraci ji snížíme o 0.1 – snižování závislosti na mimo-doménovém systému

Uvedené výsledky odpovídají adaptační sadě 10000 i-vektorů od 1000 mluvčích. Mimo doménový systém byla zastoupena 32921 i-vektory.

Pro zachování jednotnosti nebyla v testech $\eta = 0$ použita filtrace výsledků odhadu identit, protože v ostatních případech se neuplatňovala.

gender	**newmindcf**	**oldmindcf**	**EER**
	0.642	0.248	6.00%
f	0.683	0.257	6.04%
m	0.583	0.235	5.84%

Tabulka 6.4: Průměrné výsledky metody adaptace založené na shlukování a přetrénování pro hodnotu $\eta = 0$.

gender	**newmindcf**	**oldmindcf**	**EER**
	0.664	0.229	4.73%
f	0.715	0.244	4.89%
m	0.601	0.209	4.08%

Tabulka 6.5: Průměrné výsledky metody adaptace založené na shlukování a přetrénování pro hodnotu $\eta = 0.5$.

Z uvedených 4 hodnot η nejlepšího výsledku dosáhl případ s plným využitím mimo-doménové trénovací sady (výsledky v tabulkách 6.4, 6.5, 6.6 a 6.7). Tato sada nám v počátcích jistě poskytne potřebnou podporu z hlediska chyb v odhadech v adaptační sadě. To lze vidět na testech s hodnotou $\eta = 0$, kde nízký výsledek je způsoben nedostatečně přesným odhadem identit. Zároveň nám, ale mimo-doménová data výslednou variabilitu unáší svým směrem. U případu s postupným snižováním mimo-doménového podílu vidíme, že výsledek je i přes snižování chybné domény nevalný. Stará doména zřejmě zanechává v odhadech identit odchylku, která brání následnému zlepšování. Nevýhodou je zde znalost pouze přibližného poměru míchání variabilit, které kvůli způsobu míchání v trénovací sadě nemáme dostatečně pod kontrolou.

gender	newmindcf	oldmindcf	EER
	0.648	0.215	4.46%
f	0.692	0.229	4.85%
m	0.583	0.196	3.72%

Tabulka 6.6: Průměrné výsledky metody adaptace založené na shlukování a přetrénování pro hodnotu $\eta = 1$.

gender	newmindcf	oldmindcf	EER
	0.677	0.242	5.06%
f	0.731	0.257	5.42%
m	0.614	0.218	4.23%

Tabulka 6.7: Průměrné výsledky metody adaptace založené na shlukování a přetrénování pro hodnotu $\eta = 1$ s postupným poklesem.

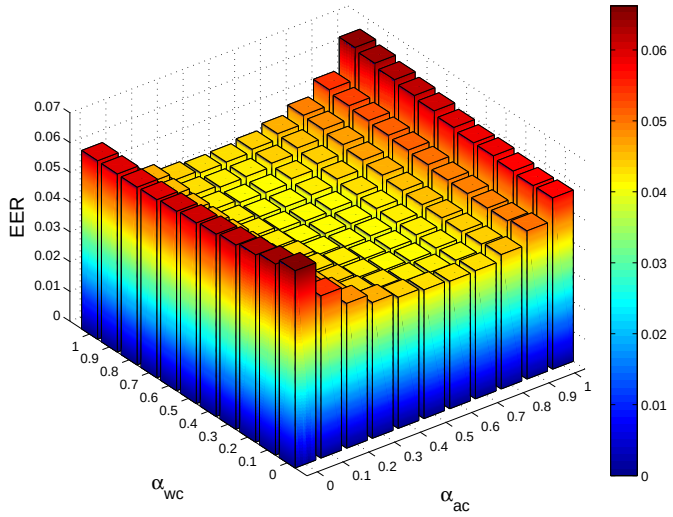
6.4.2 Shlukování a adaptace

V této metodě byla vyzdvihnuta možnost adaptovat každý koeficient (α_{ac} a α_{wc}) zvláště. Jelikož není nástroj, jenž by nám určil optimální nastavení těchto parametrů, provedeme kompletní plošné otestování jejich nastavení v rozsahu $\langle 0, 1\rangle$. Tímto dále zjistíme chování adaptace kolem hodnot s optimálním poměrem.

Pro test byla vždy provedena 1 iterace adaptačního algoritmu s danými hodnotami α. Průběhy chyby EER jsou znázorněny na obrázcích 6.3, 6.4 a 6.5.

Na snímcích vidíme, že se nejlépe systém adaptoval u mužských nahrávek. Dalo by se tedy předpokládat, že celkově se mužské nahrávky vyskytovaly v adaptační sadě více. Dále na snímcích vidíme rozsah hodnot α, kdy byla adaptace nejvíce učinná (po první iteraci). Pro obě váhovací proměnné α_{ac} a α_{wc} se minimum chyby pohybuje v rozsahu hodnot $\langle 0.4, 0.6\rangle$. Neodpovídá to vkládanému poměru trénovacích dat pro jednotlivé variability mimo-doménových a doménových dat. Přesto tento předpoklad nemůžeme úplně vyloučit, neboť nemáme dostatek informací o trénovacích a adaptačních nahrávkách. Víme sice parametry nahrávání a přenosového kanálu, ale podrobné informace o samotných mluvčích chybí.
Ze všech grafů 6.3, 6.4 a 6.5 je dále patrné, že se nám vždy vyplatilo adaptovat obě variability.

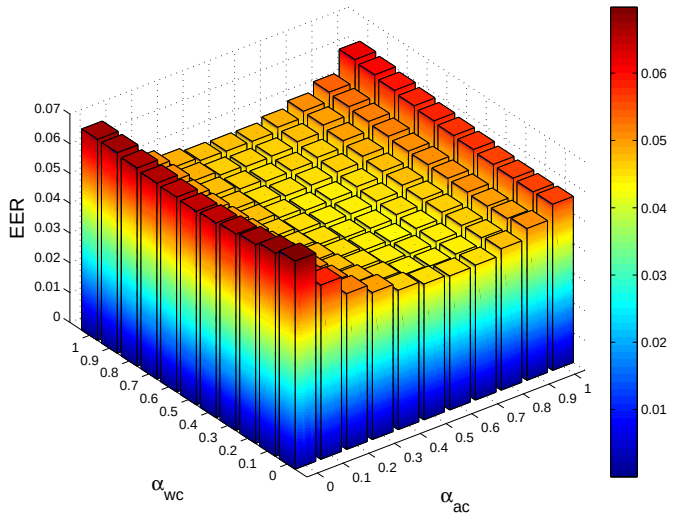
Při pozornějším pohledu dále zjistíme, že uvedené váhovací poměry budou zřejmě platné pouze v jedné iteraci. Při následující iteraci, jak je tomu uvedeno v algoritmu, vždy využíváme výsledný model z předchozí iterace jako starý systém. V každé iteraci nám tedy starý model reprezentuje jinou doménu a tudíž uvedené poměry si budou jistě mezi iteracemi také měnit. Vyplývá z toho jasná potřeba automatického nástroje, jenž by určoval optimální nastavení parametrů.



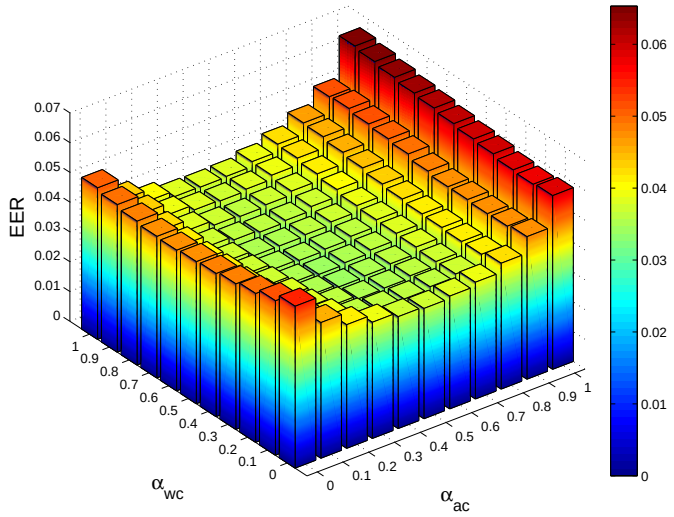
Obrázek 6.3: Ukázka vývoje EER pro jednotlivé hodnoty váhování α_{ac} a α_{wc} pro kombinovanou evaluační sadu.

gender	**newmindcf**	**oldmindcf**	**EER**
	0.581	0.175	3.61%
f	0.628	0.190	3.92%
m	0.516	0.157	3.22%

Tabulka 6.8: Průměrné výsledky metody adaptace založené na shlukování a adaptaci pro hodnotu $\alpha_{ac} = \alpha_{wc} = 0.4$.



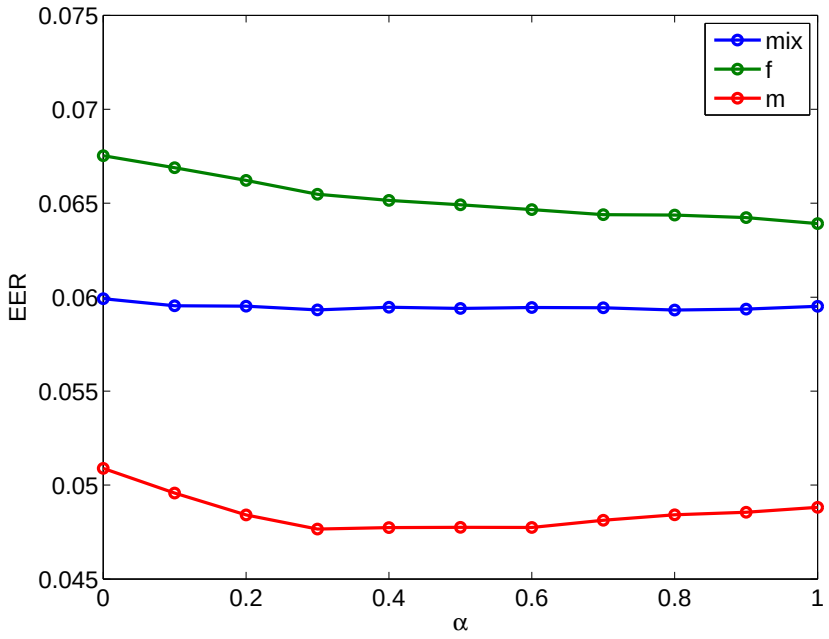
Obrázek 6.4: Ukázka vývoje EER pro jednotlivé hodnoty váhování α_{ac} a α_{wc} pro evaluační sadu žen.



Obrázek 6.5: Ukázka vývoje EER pro jednotlivé hodnoty váhování α_{ac} a α_{wc} pro evaluační sadu mužů.

6.4.3 Podvrhnutí modelu

V této technice nejde ani tak o samotnou adaptaci, jako o přímé dosazení určité variability. Očekávání od této techniky nebyla valná, přesto je nutné zmínit výhodu jejího použití. Adaptace podvrhnutím je provedena prakticky okamžitě. Vyžaduje také minimum početních úkonů v porovnání s předchozími technikami.



Obrázek 6.6: Ukázka vývoje EER pro jednotlivé hodnoty váhování α.

gender	newmindcf	oldmindcf	EER
	0.723	0.273	5.84%
f	0.760	0.292	6.53%
m	0.663	0.249	4.64%

Tabulka 6.9: Průměrné výsledky metody adaptace založené na podvrhnutí AC modelu pro hodnotu $\alpha = 0.3$.

Na snímku 6.6, lze vidět průměrný vývoj chyby EER v závislosti na hodnotě koeficientu α. V uvedeném grafu jsme záměrně uvedli vývoj všech tří hodnocených podskupin mluvčích (mix – ženy a muži dohromady, f– ženy, m– muži), neboť každá reagovala výrazněji jinak. Postupná změna chyby EER je však minimální. Projevuje se zde jistě i šířka kovariance všech dat oproti kovarianci identit mluvčích. Největší změna se projevila u datové evaluační sadě s muži. Je to zřejmě důsledek příliš malé přesnosti vstupní kovariance dat použité jako kovariance mluvčích. Dosaženého zlepšení je dosaženo spíše ustředněním evaluační sady na adaptačních datech, než samotným váhováním modelů, které přineslo minimální změnu. Celkově je výsledek systému lepší než u neadaptovaného systému. V porovnání s ostatními adaptačními technikami, je zlepšení nejnižší, což bylo očekávané. Celkové průměrné výsledky jsou uvedeny v tabulce 6.9.

6.4.4 Shrnutí porovnání adaptačních technik

Porovnání tří vytvořených a otestovaných adaptačních algoritmů dopadlo podle předpokladů. Dosažené výsledky jsou shrnuty v tabulce 6.10, kde jsou také porovnány s mimo-doménovým systémem a systémem adaptovaným s učitelem.

Nejhoršího výsledku dosáhla metoda s podvrhnutím modelu. Tento výsledek byl očekávaný. Nízká míra adaptace je způsobena tím, že upravujeme pouze jednu variabilitu modelu a to variabilitu mluvčích. Dále také vkládaná variabilita neodpovídá mluvčím, ale všem datům. Nelze tedy od této metody očekávat závratnou přesnost. Metoda má ovšem jednu výhodu, kterou je rychlost úpravy modelu, je prakticky okamžitá.

Druhého nejlepšího výsledku dosáhla metoda se shlukováním a přetrénováním. Výhodou metody je, že není nutné zasahovat do implementace systému a úprava variability probíhá na úrovni trénovací sady. Z provedených experimentů byl nejlepší případ, kdy jsme používali celou mimo-doménovou sadu, ve výsledku jsme tedy jen trénované variability rozšiřovali. To se jistě projevilo v omezení možného zlepšení výsledků, protože jsme neprováděli přechod přímo k variabilitě požadované domény. Metoda je ovšem méně náchylná na kvalitu odhadu v počátečních krocích adaptace, zároveň mísení trénovacích dat snižuje učinnost filtrace virtuálních mluvčích na nulu. Jelikož k míchání (součtu) variabilit staré a nové domény dochází již v trénovacích datech, nemáme plně pod kontrolou jejich přesný výsledný poměr. Známe sice poměr trénovacích vektorů staré ku nové doméně, to nám nezaručí, že výsledně natrénované variability v modelu budou tomuto poměru odpovídat.

Podle očekávání si nejlépe vedla metoda se shlukováním a adaptací. Výhodou metody, jak bylo řečeno, je možná adaptace každé variability zvlášť. Metoda provádí pomocí váženého průměru postupný přechod od jedné domény k druhé. Oproti předchozí metodě nám v modelu nezůstávají čistě mimo-doménové pozůstatky. Díky způsobu míchání variabilit na úrovni modelu máme možnost přesného míchání v daném poměru.

Adaptace	**EER**
systém bez adaptace	6.60%
shlukování a přetrénování	4.46%
shlukování a adaptace	3.61%
podvrhnutí modelu	5.84%
adaptace s učitelem	2.57%

Tabulka 6.10: Celkové srovnání adaptačních technik. Výsledné hodnoty EER odpovídají míchané evaluační sadě (ženy a muži dohromady). Výsledky jsou pro adaptační sadu s tisícem mluvčích.

Pro připomenutí uveďme základní princip všech adaptačních metod. Shlukování a přetrénování využívá míchání trénovací sady, shlukování a adaptace využívá váženého průměru kovariančních matic PLDA modelů a podvrhnutí variability využívá váženého průměru kovariance mluvčích mimo-doménového systému a odhadnuté kovariance mluvčích z kovariance všech dat.

Kapitola 7

Závěr

7.1 Shrnutí

Adaptace v oboru rozpoznávání mluvčích má jistě své místo a uplatnění. V rámci této práce bylo vytvořeno několik metod, ať již na samotnou adaptaci, tak na potřebné kroky použité v průběhu adaptace. U jednotlivých kroků jsme provedli měření jejich úspěšnosti a přesnosti, jak vzhledem ke konkrétní problematice (shlukování, odhad shluků), tak z pohledu užití v adaptaci a vlivu na fungování rozpoznávacího systému jako celku. Probráno tak bylo několik méně, či více známých algoritmů, jenž se ukázaly jako užitečné v další možné problematice, tedy adaptaci systémů na rozpoznání mluvčího.

Odhad identit mluvčích v adaptační sadě je klíčový krok adaptace. Pro jeho zpřesnění jsme tedy navrhli postup, jenž má za následek celkově nižší chybu celého rozpoznávacího systému. Dále jsme navrhli postup jak při adaptační proceduře rozeznat, že je čas adaptaci ukončit.

Na základě dílčích výsledků úspěšnosti jsme sestavili celkový adaptující se systém minimalizující chybu rozpoznávání. Výsledkem práce je tedy několik různých systémů adaptujících se na podmínky nové domény, minimalizující tak chybu samotného rozpoznávání. Výsledky konečného testování adaptací naznačují, že vytvořené metody by jistě našly své uplatnění při reálné aplikaci rozpoznávacích systémů.

Důležitým výsledkem práce je také získání přehledu v problematice rozpoznávání mluvčích a fungování systému používaných v této problematice.

7.2 Budoucí práce

Při práci na samotných metodách i při dílčím vyhodnocení se ukázalo, že slabým místem adaptačních technik je odhad počtu mluvčích v adaptační sadě. Pro kompenzaci chyby tohoto kroku jsme navrhli postup založený na filtraci. Přesto by jistě bylo vhodné podrobit tento krok dalšímu zkoumání a zjistit, zda výsledná chyba je způsobena metodou, či chybějící počáteční informací. Použitá metoda se zabývala odhadem v datech obecně. Lepších výsledků by zřejmě bylo možné dosáhnout navržením obdobné metody pouze pro naši problematiku odhadu počtu mluvčích.

U jednotlivých adaptačních technik jsme se zabývali spíše technikou samotnou něž kontextem jejího použití. Ze samotné podstaty problému adaptace bez učitele vyplývá potřeba dalšího nástroje, jenž by byl schopný na samotných neanotovaných datech rozhodnout, zda vůbec k adaptaci přistupovat. Myslíme tím tedy nástroj, který by byl schopný určit, zda

se trénovací a evaluační doména opravdu liší. V ideálním případě by tento nástroj měl poskytovat informaci o optimálním nastavení parametrů pro adaptaci. Docílit by se toho pravděpodobně dalo studiem rozdílnosti kovariančních matic jednotlivých datových sad.

Literatura

[1] The NIST Year 2010 Speaker Recognition Evaluation Plan. [Online], [cit. 2013-12-25]. URL http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf

[2] Speaker Recognition Evaluation. [Online], [cit. 2013-12-25]. URL http://www.itl.nist.gov/iad/mig/tests/sre/

[3] Bishop, C. M.: *Pattern Recognition and Machine Learning*. Springer, 2007, ISBN 978-0-387-31073-2.

[4] Black, P. E.: Dictionary of Algorithms and Data Structures. 1998, [Online], [rev. 2013-10-15], [cit. 2013-12-25]. URL http://xlinux.nist.gov/dads//

[5] Burget, L.; Plchot, O.; Cumani, S.; aj.: Discriminatively trained Probabilistic Linear Discriminant Analysis for speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, ISSN 1520-6149, s. 4832–4835, doi:10.1109/ICASSP.2011.5947437.

[6] Dehak, N.; Kenny, P. J.; Dehak, R.; aj.: Front-End Factor Analysis for Speaker Verification. In *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, ročník 19, 2011, s. 19–41.

[7] van Dongen, S.: *Graph Clustering by Flow Simulation*. Dizertační práce, University of Utrecht, May 2000.

[8] Everitt, B. S.; Landau, S.; Leese, M.; aj.: *Cluster Analysis*. King's College London, UK: John Wiley & Sons, Ltd, páté vydání, 2011, ISBN 978-0-470-97780-4.

[9] Fränti, P.: Number of clusters. [Online], [rev. 2013-7-2], [cit. 2013-12-25]. URL http://cs.joensuu.fi/pages/franti/cluster/Clustering-Part3.ppt

[10] Garcia-Romero, D.; McCree, A.: SUPERVISED DOMAIN ADAPTATION FOR I-VECTOR BASED SPEAKER RECOGNITION. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, s. 4075–4079.

[11] Glembek, O.: *Optimalization of Gaussian Mixture Syubspace Models and Related Scoring Algorithms in Speaker Verification*. Dizertační práce, Brno University of Technology, Faculty of Information Technology, Brno, 2012.

[12] Gradshteyn, I.; Ryzhik, I.: *Table of Integrals, Series, and Products*. Academic Press, 7 vydání, 2007, ISBN 978-0-12-373637-6.

[13] Manning, C. D.; Raghavan, P.; Schütze, H.: *An Introduction to Information Retrieval*. Cambridge University Press, April 2009, ISBN 0521865719.

[14] Niko, B.; Lukáš, B.; Patrick, K.; aj.: ABC System description for NIST SRE 2010. In *Proc. NIST 2010 Speaker Recognition Evaluation*, Brno, CZ, 2010, s. 1–20.

[15] Pham, D. T.; Dimov, S. S.; Nguyen, C. D.: Selection of K in K-means clustering. In *Journal of Mechanical Engineering Science*, ročník 219, 2005, s. 103–119.

[16] Prince, S. J.; Elder, J. H.: Probabilistic Linear Discriminant Analysis for Inferences About Identity. In *IEEE International Conference on Computer Vision*, 2007.

[17] Reynolds, D. A.; Quatieri, T. F.; Dunn, R. B.: Speaker Verification Using Adapted Gaussian Mixture Models. In *Digital Signal Processing*, 2000, s. 19–41.

[18] van Rijsbergen, C. J.: *Information Retrieval*. London: Butterworths, druhé vydání, 1979, [Online], [cit. 2013-12-25].
URL http://www.dcs.gla.ac.uk/Keith/Preface.html

[19] Satopää, V.; Albrecht, J.; Irwin, D.; aj.: Finding a „Kneedle“ in a Haystack:Detecting Knee Points in System Behavior. [Online], [cit. 2013-12-25].
URL http://www1.icsi.berkeley.edu/~barath/papers/kneedle-simplex11.pdf

[20] Silovský, J.: *Generativní a dikrimiativní klasifikátory v úlohách textově nezávislého rozpoznávání a diarizace mluvších*. Dizertační práce, Technická univerzita v Liberci, 2011.

[21] Tan, P.-N.; Steinbach, M.; Kumar, V.: *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., první vydání, 2005, ISBN 0321321367.

[22] Vaquero, C.: Adaptation to New Domains, Quality Measures for Clustering. 2013.

[23] Xu, M.; Duan, L.-Y.; Cai, J.; aj.: HMM-Based Audio Keyword Generation. In *Advances in Multimedia Information Processing - PCM 2004*, ročník 333, 2005, ISBN 978-3-540-30543-9, s. 566–574.

[24] Zhao; Karypis: Cluster Evaluation. 2002, [Online], [cit. 2013-12-25].
URL https://facwiki.cs.byu.edu/nlp/images/2/25/ClusterEvaluation.ppt

[25] Zhao, Q.; Hautamaki, V.; Fränti, P.: Knee Point Detection in BIC for Detecting the Number of Clusters. In *Advanced Concepts for Intelligent Vision Systems*, ročník 5259, Springer Berlin Heidelberg, 2008, ISBN 978-3-540-88458-3, ISSN 0302-9743, s. 664–673.
URL http://link.springer.com/chapter/10.1007/978-3-540-88458-3_60

[26] Řezanková, H.; Húsek, D.; Snášel, V.: *Shluková analýza dat*. Professional publishing, druhé vydání, 2009, ISBN 978-80-86946-81-8.

Dodatek A

Obsah CD

- `DP.pdf` – text této práce
- `doc/` – zdrojové kódy dokumentace
- `projekt/` – zdrojové kódy v Matlabu a přislušné testy
  - `baseline/` – základní skóre
  - `pomocne_skripty/` – skripty s dílčími kroky adaptace (shlukovací algoritmy, odhadu počtu shluků, filtrace ...)
  - `system/` – skripty jednotlivých adaptačních metod a PLDA
    - `adaptacni_fce/` – jednotlivé adaptační funkce
    - `adaptacni_fce_experimentalni/` – experimentální adaptační funkce (Stejné výsledky jako `adaptacni_fce`, ale počítají kvalitu shlukování, filtrace ... (Byli použity k testům a měřením.)
    - `cov_plda/` – dvou-kovarianční PLDA (trénování, převod z obecného PLDA, skórování podobnosti ...)
    - `plda_adapting/` – obecné PLDA **Speech@FIT** s upraveným rozhraním pro možnou manuální inicializaci matic U,V a D
    - `plda_train/` – jednoduché a dvojité (rozdílné trénovací sady pro U a V,D) trénování PLDA
    - `run_plda_training_adapt_pure.m` – upravený skript `run_plda_training.m` skupiny **Speech@FIT** s vloženou adaptační funkcí
    - `run_plda_training_adapt_v1.m` – upravený skript `run_plda_training.m` skupiny **Speech@FIT** s vloženou adaptační funkcí (s užitím experimentálních verzí adaptace)
    - `SRE_plda_adaptation_v1.sh` – spouštěcí skript testů adaptace, využívá definovaných cest k i-vektorům, evaluačním listům atd.
  - `testy` – testy jednotlivých částí adaptace a kompletních metod
    - `adaptacni_metody/` – testy sestavených adaptačních metod
    - `casti/` – testy jednotlivých kroků adaptace (shlukovacích algoritmů, odhadu počtu shluků, filtrace ...)
      - `filtrace/` – testy filtrace a jejím vlivu na systém
      - `odhad_poctu/` – testy odhadu počtu mluvčích

    - `shlukovani/` – testy shlukovacich algoritmu
    - `vyber_shluk/` – testy chybovosti systému v závislosti na shlukovacím algoritmu
    - `vyber_vzdalenosti/` – testy vhodné měřené vzdálenosti (Euklidové, kosínové)
  - `oprimalni_alfa/` – testy optimálního nastavení koeficientu α, α_{ac} a α_{wc}.
  - `stop_kriterium/` – testy zastavovacího kritéria adaptace

V jednotlivých složkách s testy jsou vždy vloženy testovací skripty a dosažené výsledky. V případě, že složky neobsahují testovací skripty, byli k daným výsledkům využity přímo adaptační funkce, nebo dílčí kroky adaptace.

Dodatek B

Kompletní výsledky testů

B.1 Výběr shlukovacího algoritmu

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.645	0.231	5.18%	7169	408950
f	0.684	0.241	5.30%	3704	233077
m	0.586	0.218	4.91%	3465	175873

Tabulka B.1: Výsledky systému se Spojováním komponent.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.670	0.245	5.48%	7169	408950
f	0.707	0.256	5.66%	3704	233077
m	0.612	0.230	5.18%	3465	175873

Tabulka B.2: Výsledky systému s Aglomerativní hierarchickým shlukováním.

B.2 Vliv filtrace na chybu systému

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.642	0.248	6.00%	7169	408950
f	0.684	0.257	6.04%	3704	233077
m	0.583	0.235	5.83%	3465	175873

Tabulka B.3: Výsledky systému s jednou iterací adaptace bez filtrování výsledků shlukování.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.707	0.277	6.30%	7169	408950
f	0.742	0.286	6.30%	3704	233077
m	0.638	0.262	6.20%	3465	175873

Tabulka B.4: Výsledky systému s jednou iterací adaptace a výběrem virtuálních mluvčích s přesně 10 i-vektory.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.645	0.232	5.18%	7169	408950
f	0.684	0.241	5.30%	3704	233077
m	0.587	0.218	4.91%	3465	175873

Tabulka B.5: Výsledky systému s jednou iterací adaptace a výběrem virtuálních mluvčích s minimálně 5 a maximálně 15 i-vektory.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.642	0.220.	4.73%	7169	408950
f	0.682	0.230	5.00%	3704	233077
m	0.580	0.202	4.28%	3465	175873

Tabulka B.6: Výsledky systému s jednou iterací adaptace a výběrem virtuálních mluvčích s minimálně 5 a maximálně 15 i-vektory při správném počtu mluvčích.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.643	0.228	5.10%	7169	408950
f	0.681	0.237	5.20%	3704	233077
m	0.581	0.213	4.78%	3465	175873

Tabulka B.7: Výsledky systému s jednou iterací adaptace a výběrem virtuálních mluvčích s minimálně 5 a maximálně 15 i-vektory pro trénování AC a s minimálně 3 a maximálně 20 pro trénování WC.

B.3 Výsledky jednotlivých adaptačních metod

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.687	0.282	6.84%	7169	408950
f	0.716	0.289	6.84%	3704	233077
m	0.641	0.268	6.64%	3465	175873

Tabulka B.8: Výsledky adaptace s přetrénováním pro 500 mluvčích v adaptační sadě a $\eta = 0$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.700	0.254	5.32%	7169	408950
f	0.747	0.266	5.68%	3704	233077
m	0.642	0.234	4.54%	3465	175873

Tabulka B.9: Výsledky adaptace s přetrénováním pro 500 mluvčích v adaptační sadě a $\eta = 0.5$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.663	0.229	4.77%	7169	408950
f	0.709	0.241	5.19%	3704	233077
m	0.600	0.210	3.97%	3465	175873

Tabulka B.10: Výsledky adaptace s přetrénováním pro 500 mluvčích v adaptační sadě a $\eta = 1$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.714	0.269	5.80%	7169	408950
f	0.757	0.285	6.24%	3704	233077
m	0.655	0.246	4.85%	3465	175873

Tabulka B.11: Výsledky adaptace s přetrénováním pro 500 mluvčích v adaptační sadě a $\eta = 1$ s poklesem o 10% v každé iteraci.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.642	0.248	6.00%	7169	408950
f	0.683	0.257	6.04%	3704	233077
m	0.583	0.235	5.84%	3465	175873

Tabulka B.12: Výsledky adaptace s přetrénováním pro 1000 mluvčích v adaptační sadě a $\eta = 0$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.664	0.229	4.73%	7169	408950
f	0.715	0.244	4.89%	3704	233077
m	0.601	0.209	4.08%	3465	175873

Tabulka B.13: Výsledky adaptace s přetrénováním pro 1000 mluvčích v adaptační sadě a $\eta = 0.5$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.648	0.215	4.46%	7169	408950
f	0.692	0.229	4.85%	3704	233077
m	0.583	0.196	3.72%	3465	175873

Tabulka B.14: Výsledky adaptace s přetrénováním pro 1000 mluvčích v adaptační sadě a $\eta = 1$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.677	0.242	5.06%	7169	408950
f	0.731	0.257	5.42%	3704	233077
m	0.614	0.218	4.23%	3465	175873

Tabulka B.15: Výsledky adaptace s přetrénováním pro 1000 mluvčích v adaptační sadě a $\eta = 1$ s poklesem o 10% v každé iteraci.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.600	0.193	4.05%	7169	408950
f	0.643	0.206	4.30%	3704	233077
m	0.539	0.176	3.70%	3465	175873

Tabulka B.16: Výsledky adaptace s váhováním modelů pro 500 mluvčích v adaptační sadě a $\alpha_{ac} = \alpha_{aw} = 0.4$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.581	0.175	3.61%	7169	408950
f	0.628	0.190	3.92%	3704	233077
m	0.516	0.157	3.22%	3465	175873

Tabulka B.17: Výsledky adaptace s váhováním modelů pro 1000 mluvčích v adaptační sadě a $\alpha_{ac} = \alpha_{aw} = 0.4$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.723	0.273	5.84%	7169	408950
f	0.760	0.292	6.53%	3704	233077
m	0.663	0.249	4.64%	3465	175873

Tabulka B.18: Výsledky adaptace s podvrhnutím modelu pro 1000 mluvčích v adaptační sadě a $\alpha = 0.3$.

gender	newmindcf	oldmindcf	EER	targets	nontargets
	0.716	0.280	6.11%	7169	408950
f	0.755	0.296	6.64%	3704	233077
m	0.665	0.259	4.89%	3465	175873

Tabulka B.19: Výsledky adaptace s podvrhnutím modelu pro 500 mluvčích v adaptační sadě a $\alpha = 0.3$.