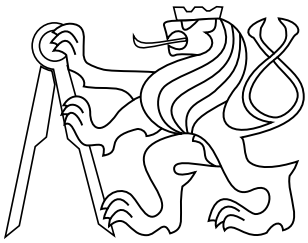
Diplomová práce

Moderní metody rozpoznávání mluvčího na bázi GMM a DNN

Michael Záruba



leden 2017

Vedoucí práce: doc. Ing. Petr Pollák, CSc.

České vysoké učení technické v Praze
Fakulta elektrotechnická, Katedra teorie obvodů

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra radioelektroniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Michael Záruba**

Studijní program: Komunikace, multimédia a elektronika
Obor: Komunikační systémy

Název tématu: **Moderní metody rozpoznávání mluvčího na bázi GMM a DNN**

Pokyny pro vypracování:

1. Seznamte se s problematikou rozpoznávání mluvčího se zaměřením na metody na bázi GMM a DNN. Proveďte přehledovou rešerši aktuálně nejpřesnějších a nejvíce používaných technik. Zvláštní pozornost věnujte také výběru vhodných řečových příznaků pro rozpoznávání.
2. Vybrané metody implementujte s nástroji KALDI. Na základě zavedených konvencí vytvořte pro danou úlohu skripty ("recepty") umožňující jejich využití na řešitelském pracovišti resp. sdílení odbornou komunitou v oblasti rozpoznávání řeči.
3. V experimentální části ověřte funkčnost implementovaných metod a vyhodnoťte přesnost rozpoznávání mluvčího na dostupných databázích mluvené řeči. Dosažené výsledky srovnejte s výsledky dosahovanými na analogických úlohách jinými autory.

Seznam odborné literatury:

[1] J. Psutka, L. Müller, J. Matoušek, V. Radová. Mluvíme s počítačem česky. Academia 2006.
[2] X. Huang, A. Acero, H.-W. Hon. Spoken Language Processing. Prentice Hall, 2001.
[3] D. Povey et al, The Kaldi Speech Recognition Toolkit. In Proc. of IEEE 2011 ASRU, Hawai, US, 2011. Note. Project WEB-page http://kaldi.sourceforge.net/.
[4] F. Richardson, D. Reynolds, N. Dehak: Deep Neural Network Approaches to Speaker and Language Recognition. IEEE Signal Processing Letters, Vol. 22, No. 10, October 2015.
[5] M. McLaren, Y. Lei, L. Ferrer: Advances in Deep Neural Network Approaches to Speaker Recognition. In Proc. of ICASSP 2015, Brisbane, Australia, pp. 4814-4818.
[6] T. Kinnunen, H. Li: An overview of text-independent speaker recognition: From features to supervectors. Speech Communication Vol. 52, 2010, pp. 12-40.

Vedoucí: doc. Petr Pollák Ing., CSc.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

doc. Mgr. Petr Páta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 18. 2. 2016

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne: .....

Podpis: .....

Poděkování

Rád bych poděkoval panu doc. Ing. Petru Pollákovi, CSc., za odborné vedení a cenné rady, které mi pomohly při psaní této práce. Můj dík patří také mé manželce za její péči.

Abstrakt

Tato práce se zabývá úlohou hlasové identifikace a verifikace mluvčího. Jejím hlavním cílem je popsat v současné době nejpoužívanější postupy a vybrané metody experimentálně ověřit na dostupných datech. Hlavní pozornost je věnovaná především statistickému modelování na bázi GMM, respektive reprezentaci mluvčích založené na i-vektorech. Dále se práce zaměřuje na metody zvýšení přesnosti identifikace na bázi lineární diskriminační analýzy (LDA), respektive pravděpodobnostní lineární diskriminační analýzy (PLDA).

V praktické části byly výše popsané metody realizovány pomocí nástrojů sady KALDI. Přesnost identifikace a verifikace byla otestována na dvou českých databázích, SPEECON a SpeechDat, a to pro různé vstupní podmínky a parametry úlohy. Během realizace byly vytvořeny skripty („recepty“) v souladu se zavedenými standardy sady KALDI.

Nejlepších výsledků bylo dosaženo u databáze SPEECON, u které se podařilo pomocí metody PLDA bezchybně identifikovat a verifikovat 284 rozpoznávaných mluvčích. Implementace může být případně později rozšířena za účelem otestování dalších parametrů úlohy nebo jiných databází. Výsledky předložené práce mohou také posloužit při implementaci úlohy hlasové identifikace či verifikace řečníka v reálném provozu.

Klíčová slova

rozpoznávání mluvčího; identifikace mluvčího; verifikace mluvčího; GMM; UBM; i-vektor; KALDI

Abstract

The present thesis describes voice-based speaker identification and verification, and its main objective is to describe currently the most frequently used techniques and to realize experiments with selected methods using available speech data. The main attention is paied to statistical modelling based on GMM and the representation of speakers based on i-vectors. Further, the attention is turned to methods used to increase the precision of identification, i.e. techniques based on linear discriminant analysis (LDA) or probabilistic linear discriminant analysis (PLDA).

In the practical part, described methods were implemented using the KALDI toolkit, and the accuracy of identification and verification was tested for various input conditions and algorithm setups. Two available Czech speech databases, SPEECON and SpeechDat, were used for realized experiments. Throughout the implementation, scripts ("recipes") were created in accordance with approved standards of the KALDI toolkit.

The best results were obtained for the PLDA method and SPEECON speech data, where 284 tested speakers were successfully identified and verified with zero error. The implementation may be expanded, with the purpose to include the testing of other algorithms or used speech databases. The results of presented thesis may contribute the implementation of voice-based speaker identification or verification within a real application.

Keywords

speaker recognition; identification of speaker; speaker verification; GMM; UBM; i-vector; KALDI

Obsah

Přílohy

Zkratky

Seznam zkratek používaných v textu.

BASH	interpret příkazového řádku (Bourne Again SHell)
DCF	DCF funkce (Detection Cost Function)
DCT	diskrétní kosinová transformace (Discrete Cosine Transform)
DFT	diskrétní Fourierova transformace (Discrete Fourier Transform)
DNN	hluboké umělé neuronové sítě (Deep Neural Network)
EER	míra stejné chyby (Equal Error Rate)
EM	očekávaná maximalizace (Expectation Maximization)
FAR	míra chybného odmítnutí (False Acceptance Rate)
FFT	rychlá Fourierova transformace (Fast Fourier Transform)
FRR	míra chybného přijetí (False Rejection Rate)
GMM	model Gaussových hustotních směsí (Gaussian Mixture Model)
IDFT	inverzní diskrétní Fourierova transformace (Inversion Discrete Fourier Transform)
LDA	lineární diskriminativní analýza (Linear Discriminant Analysis)
MAP	maximální aposteriorní pravděpodobnost (Maximum A Posteriori)
MFCC	mel-frekvenční kepstrální koeficienty (Mel-Frequency Cepstral Coefficients)
NIST	Americký úřad pro standardizaci a technologie (National Institute of Standards and Technology)
OSS	software s otevřeným zdrojovým kódem (Open-Source Software)
PCM	pulzně kódová modulace (Pulse Code Modulation)
PLDA	pravděpodobnostní lineární diskriminativní analýza (Probabilistic Linear Discriminant Analysis)
ROC	ROC křivka (Receiver Operating Characteristics)
SGE	systém pro správu dávkového zpracování úloh (Sun Grid Engine)
UBM	univerzální model pozadí (Universal Background Model)
VAD	detekce hlasové aktivity (Voice Activity Detection)

1 Úvod

Úloha rozpoznávání mluvčího je jednou z mnoha technických disciplín v oblasti analýzy a syntézy hlasu. Toto odvětví zaznamenalo velký rozvoj s příchodem výpočetní techniky. I v dřívějších dobách mělo lidstvo touhu strojově rozpoznávat a vytvářet řeč, ale prostředky k realizaci byly velmi omezené. Objevovaly se například pokusy realizovat syntézu hlasu mechanicky. Zatímco u syntézy hlasu je klíčová snaha transformovat textový zápis na audio signál, naproti tomu analýza hlasu má více disciplín, nikoliv pouze převod audio signálu řeči do textové podoby.

Důležitou technickou disciplínou je rozpoznávání jazyka a také mluvčího. Tyto dvě zmíněné disciplíny mají svá uplatnění buď jako podpůrné systémy, anebo jako samostatně pracující systémy. Automatický rozpoznávač řeči bude pracovat přesněji, pokud bude znát charakteristiku řeči konkrétního mluvčího. Pracuje-li se systémem více osob, je vhodné jejich identitu nejprve zjistit, to lze provést například analýzou hlasu v krátké promluvě, k níž mluvčího vyzval systém. Samostatně pracující systémy mohou mj. přiřadit konkrétní hlasový vzorek podezřelého člověka ke konkrétní identitě v databázi natrénovaných hlasových profilů.

I když rozpoznávání mluvčího patří v porovnání s jinými variantami biometrické identifikace člověka mezi méně přesné rozpoznávací metody, v poslední době došlo v této oblasti k výraznému zvýšení přesnosti rozpoznávání. Je tomu tak zejména proto, že v posledních desetiletích silně pokročil vývoj technologií pro zpracování a ukládání akustických signálů. Výkon výpočetních prostředků pro zpracování signálů se zvyšuje exponenciálním tempem. Dalším důležitým faktorem rozvoje v oblasti rozpoznávání mluvčího je nepochybně rozsáhlá telefonní sít, která poskytuje široký záběr pro použití těchto technologií.

Možnou aplikací identifikace mluvčího je mj. doplňková kontrola identity volajícího zákazníka na linku technické podpory. Využití se nabízí například v situaci, kdy je ze strany technické podpory nutné provést technický úkon, který vyžaduje vyšší úroveň autentizace. Zákazníka technické podpory, který je již autentizován pomocí telefonního čísla volajícího, je možné ještě dodatečně analyzovat podle hlasu. Předpokladem je samozřejmě uložená charakteristika hlasu v lokální databázi mluvčích.

Další možnou aplikací identifikace s následnou verifikací je rozpoznání a ověření totožnosti osob zaregistrovaných například na konferenci. Po krátké promluvě právě identifikované osoby bude její hlas analyzován, bude vyhledána nejpodobnější charakteristika řeči a porovnána míra podobnosti. V případě neúspěšné identifikace nebo verifikace je nutné osobu autentizovat jiným způsobem. Kontrolní lidský faktor při samotné identifikaci je ale důležitý i z bezpečnostního hlediska. Jednak potvrdí shodu například vysloveného jména se jménem, které bylo identifikováno, ale také by měl odhalit případný pokus o podvod, pokud se návštěvník pokusí použít cizí nahrávku.

Využití těchto technologií se nachází i v domácím prostředí. Pokud bude v obytném prostoru nainstalován inteligentní systém pro ovládaní osvětlení, hudby a například teplotní regulace, je možné po identifikaci příchozí osoby aktivovat její přednastavený profil a vhodně nastavit úroveň osvětlení, dále ozvučit prostor oblíbenou hudbou rezidenta a zajistit pro něj tepelnou pohodu.

Tato práce popisuje v kapitole 2 nejprve problematiku rozpoznávání mluvčího obecně, dále pak analýzu signálu a hodnocení přesnosti systému. V kapitole 3 vysvětluje teoretické principy, na kterých je založena samotná klasifikace mluvčích. V další kapitole číslo 4 je nejprve čtenář obecně seznámen s nástroji KALDI, poté je v jednotlivých krocích popsána samotná implementace úlohy rozpoznávání mluvčího. Konečně kapitola 5 prezentuje výsledky experimentů pro různé vstupní podmínky úlohy.

2 Obecné principy rozpoznávání mluvčího

Problematika rozpoznávání mluvčího je založena na hledání takové informace v akustickém signálu mluvčích, která bude unikátní pro každého z nich. Pro každého mluvčího tedy bude nutné najít takovou charakteristiku, která umožní najít odlišnosti mezi jednotlivými mluvčími. Lidský hlas lze charakterizovat:

- fyzikální (vnitřní) charakteristikou řeči,
- naučenou charakteristikou řeči.

Fyzikální charakteristika je závislá na vnitřním uspořádání a anatomické stavbě lidských orgánů, jako jsou hlasivky, ústní a nosní dutina. Tuto charakteristiku nelze napodobit jiným člověkem, ale může se měnit například při nachlazení dotyčné osoby. Vyjadřuje ji mj. průběh základní frekvence hlasu a dále hodnoty frekvencí formantů a jejich šířky pásma. U jedince se mění nejvíce v jeho vývojovém období.

Naučená charakteristika popisuje ovládání řečového ústrojí z lidského mozku. I tato charakteristika se během života jedince částečně mění, ale lze ji snadněji napodobit. Lze si představit například imitátora, který bude napodobovat jiného známého mluvčího. V aplikacích, kde je potřeba ověřit totožnost mluvčího, je tedy zajímavější fyzikální charakteristika. Fyzikální charakteristiku modelují příznaky řeči založené například na kepstrální analýze.

2.1 Rozdělení metod rozpoznávání mluvčího

Rozpoznávání mluvčího lze rozdělit do dvou skupin, a to na metody se subjektivním nebo objektivním hodnocením podobnosti hlasu [1]. Hodnotiteli subjektivních metod jsou obvykle experti v oblasti fonetiky a lingvistiky. Subjektivní metody identifikace mluvčího jsou takové, které nelze jednoznačně matematicky popsat a u nichž nelze zajistit stejnou kvalitu rozpoznávání i v dlouhodobějším časovém horizontu.

Do skupiny objektivních metod bych zařadil výpočetní metody, které jsou založeny na přesných matematických postupech. Metody jsou implementovány v automatizovaném systému, který zpracuje akustický signál a provádí samotnou identifikaci, ale rozpoznávání se musí nejprve naučit, tedy natrénovat.

2.1.1 Subjektivní metody

Do skupiny subjektivních metod můžeme zařadit [1]:

- forenzní lingvistiku,
- poslechové metody,
- spektrografické metody.

Forenzní lingvistika zkoumá mluvený nebo také psaný projev jedince z hlediska používaného jazyka. Zkoumá se slovní zásoba, stavba vět, četnost používaných slov a případně chyb. Dále lze zkoumat artikulační specifika jedince a případně různé poruchy řeči.

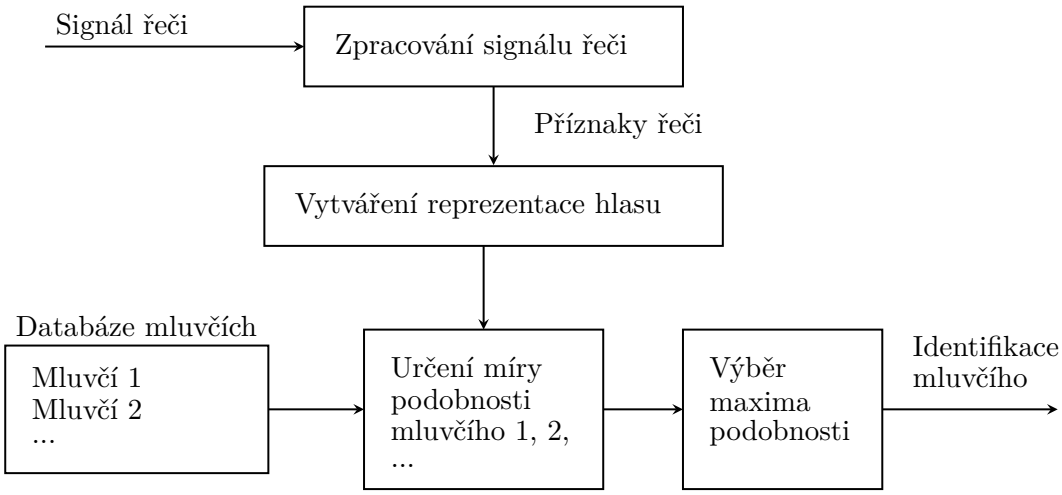
U poslechových metod skupina posluchačů subjektivně rozhoduje, zda referenční mluvčí v nahrávce je shodný s mluvčím v nahrávce rozpoznávané. Zajímavé je, že lidský hodnotitel má poměrně vysokou rozlišovací schopnost i při nízké kvalitě nahrávek.

Spektrografické metody jsou založeny na převedení zvukového signálu do grafického vyjádření spektra. Časová osa nahrávky zůstává, ale krom vynesení intenzity signálu, obvykle v podání barevné informace, je vynesena ještě informace o frekvenci signálu. Zkoumáním těchto grafů lze například podle základního tónu mluvčího a frekvencí formantů rozeznávat pouhým okem, zda byl mluvčí muž nebo žena.

2.1.2 Objektivní metody

Identifikaci mluvčího lze dle [2] provádět buď v uzavřené množině, nebo v otevřené množině. Blokové schéma na obr. 1 ukazuje, na jakém principu je založena identifikace mluvčího v množině uzavřené. Předpokládá se, že rozpoznávaný mluvčí má uloženou reprezentaci svého hlasu v databázi. Nejprve se ze signálu nahrávek extrahují řečové příznaky, které dostatečně charakterizují konkrétního mluvčího. Dále je z rozložení příznaků vytvořena například statistická reprezentace hlasu, tedy pravděpodobnostní model, který modeluje jedince. Tento model je uložen v databázi pro každého mluvčího a je porovnáván s právě rozpoznávaným mluvčím. Identifikace je realizována nalezením nejvyššího skóre podobnosti modelů.

U identifikace v otevřené množině nemusí charakteristika rozpoznávaného mluvčího být uložena v databázi. Kromě nalezení maxima, což znamená nalezení nejpodobnějšího hlasu, se provádí ještě porovnání s prahem, tedy verifikační fáze. S určitou pravděpodobností se tedy detekuje, zda nalezený model v databázi patří mluvčímu nebo nikoli. Pokud hodnota skóre nalezeného maxima je větší než stanovený práh, systém vyhodnotí mluvčího jako právě identifikovaného referenčního mluvčího. Pokud bude hodnota skóre naopak menší, je mluvčí považován za cizího[1]. Porovnávání s prahem bude vysvětleno v sekci 2.3.



Obr. 1 Identifikace mluvčího v uzavřené množině

Rozpoznávání mluvčího může být buď textově závislé, částečně textově závislé nebo textově nezávislé [2]. Textově závislé rozpoznávání je založeno na porovnávání nahrávek se stejným slovním obsahem. Obsahem může být nějaké heslo, posloupnost čísel a podobně. U autentizačních systémů může být například vygenerována náhodná posloupnost čísel a ověřovaná osoba je při verifikaci požádána o zopakování této posloupnosti. V tomto textu bude řešena problematika textově nezávislého rozpoznávání mluvčího.

[1] Někdy je cizí mluvčí označován jako podvodník (impostor).

U těchto systémů je nutné vytvořit co nejpřesnější a nejobecnější model pro všechny mluvčí, což je výpočetně velmi náročné.

2.2 Akustická analýza

2.2.1 Časová a frekvenční reprezentace signálu

Akustický signál promluv může být uložen v podobě digitálních zvukových souborů a přímo zpracován aplikací pro rozpoznání mluvčího. Tato aplikace provede předzpracování signálu v následujících krocích: preemfáze a krátkodobá spektrální analýza signálu. Pokud je automatické rozpoznávání mluvčího implementováno do systému s mikrofonem, kde je obvykle požadováno rozpoznávání v reálném čase, je nutné analogový signál nejprve digitalizovat pomocí PCM. Pulsně kódová modulace je prováděna ve dvou krocích:

- vzorkování spojitého signálu v čase,
- kvantizace a kódování hodnoty vzorku.

Při vzorkování je nutné dodržet Shannonův (Nyquistův) vzorkovací teorém. Před A/D převodník se zpravidla zapojí analogový filtr typu dolno-frekvenční propust, aby nedocházelo k problémům, jako je například aliasing.

U kvantizace, při aproximaci hodnoty vzorku na nejbližší hodnotu z konečného počtu číselných hodnot, dochází ke ztrátě informace a tato ztráta informace se projeví jako kvantizační šum. U kvantizace na nižší počet bitů je snaha používat nelineární rozložení úrovní v napěťovém rozsahu hodnot A/D převodníku, jelikož lidské ucho má lepší rozlišovací schopnost v oblasti malých hodnot.

V této práci byla úloha rozpoznávání mluvčího implementována a testována s daty dvojího typu:

- Telefonní data vzorkovaná frekvencí 8 kHz s nelineární osmibitovou kvantizací na bázi a-law.
- Širokopásmová data vzorkovaná frekvencí 16 kHz s lineární kvantizací a hloubkou 16 bitů.

Před dalším zpracováním signálu se obvykle provádí proces preemfáze. Frekvenční charakteristika lidského hlasu vykazuje s rostoucí frekvencí pokles úrovně amplitudy signálu. Pro kompenzaci tohoto jevu je vhodné pomocí preemfáze zvýšit úroveň amplitud vyšších frekvenčních složek signálu. V procesu preemfáze se používá filtr typu horní propust.

Signál upravený pomocí preemfáze je dále rozdělen pomocí váhovacího okna do krátkých kvazistacionárních časových úseků, na kterých je dále prováděna spektrální nebo kepstrální analýza. Signál je v pravidelných časových posuvech, dlouhých zpravidla 15–30 ms, násoben v časové oblasti váhovacím oknem. Posun okna se provádí o N nebo méně vzorků (N je délka okna) v rámci signálu a z celého signálu zůstává vždy pouze krátký časový úsek délky okna, zbytek signálu je vynulován. Signál z každého okna je analyzován zvlášť podle typu použitých příznaků řeči.

Nejčastěji je pro spektrální analýzu používáno Hammingovo okno[2]. Hammingovo okno si klade za cíl co nejlépe potlačit postranní laloky v modulu spektra (prosakování spektra), ale proti základnímu obdélníkovému oknu má ve spektru menší strmost. Pokud je použito Hammingovo okno, je vhodné, aby se sousední časové úseky překrývaly. Překryv je zpravidla padesátiprocentní, nebo větší.

[2]Nástroje KALDI používají jako výchozí okno Povey (jméno vývojáře), jehož parametry jsou prakticky totožné s oknem Hammingovým.

2.2.2 Řečové příznaky MFCC

Řečové příznaky jsou nejčastěji počítány na bázi kepstrální analýzy, nejčastější variantou jsou MFCC (Mel Frequency Cepstral Coefficient). Kepstrum modeluje odezvu hlasového ústrojí na sled impulsů (znělé hlásky) a na šum (neznělé hlásky). Klasické reálné kepstrum se počítá převedením signálu z časové oblasti do spektrální, následně se z absolutní hodnoty spektra provede logaritmus a zpětnou Fourierovou transformací se vypočte reálné kepstrum. U MFCC je navíc použita úprava frekvenčních pásem ve spektrální oblasti bankou mel-frekvenčních filtrů s nelineární Melovskou stupnicí [1]. Nejdůležitější informace poskytuje několik málo prvních koeficientů, jejich hodnota rychle ubývá k nule. MFCC se získávají z krátkých časových úseků, které jsou časově invariantní (neměnné). Každý kvazistacionární časový úsek signálu je zpracován postupně v následujících krocích:

1. preemfáze
2. aplikace váhovacího okna v časové oblasti
3. FFT
4. aplikace banky mel filtrů
5. logaritmus výstupního signálu z každého z N filtrů
6. DCT

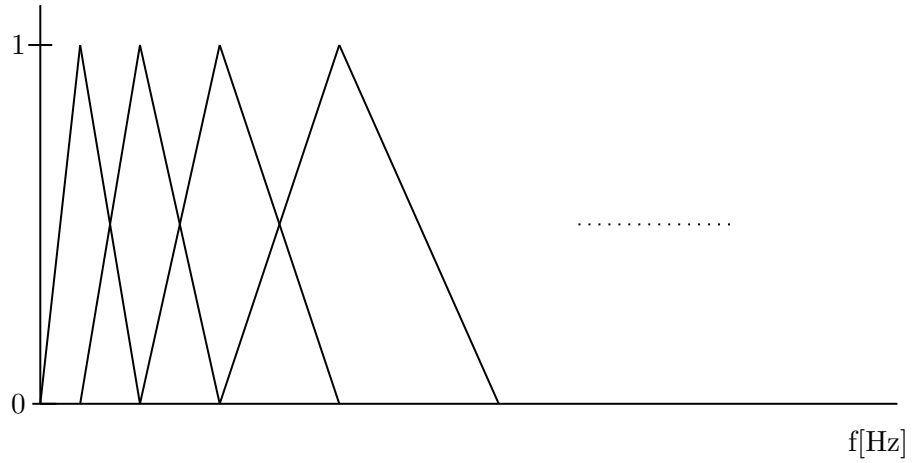
První dva kroky (preemfáze a váhování) jsou popsány v sekci 2.2.1. Fourierova transformace (FFT) předzpracovaný časový úsek převede do frekvenční oblasti, kde je spektrum váženo mel filtry. Přenosové funkce jednotlivých filtrů jsou naznačeny na obr. 2. Každý filtr je tvaru trojúhelníku a chová se jako pásmová propust. Trojúhelníkové filtry jsou navrženy tak, aby co nejlépe kompenzovaly nelineární vnímání frekvence zvuku, podobně jako lidské ucho. Dle [2] byla stupnice mel frekvencí navržena a definována tak, jak je uvedeno ve vztahu (1). Frekvence v klasické lineární stupnici je označena $f[Hz]$, frekvence s upravenou nelineární stupnicí je označena $f_m[mel]$ a je zde použit dekadický logaritmus.

$$f_m = 2595 \log\left(1 + \frac{f}{100}\right) \tag{1}$$

Po aplikaci kaskády mel-frekvenčních filtrů ve frekvenční oblasti dostáváme jako jejich výstup signály, které označíme $y_m(i)$, kde index i značí i-tý filtr. Poslední dva kroky (viz seznam výše) jsou popsány matematicky vztahem (2), kde N značí počet mel-frekvenčních filtrů a M je počet MFCC koeficientů. Výsledkem je tedy několik koeficientů $c_m(k)$, které charakterizují krátký úsek právě zpracovávaného signálu.

$$c_m(k) = \sum_{i=1}^{N} \log y_m(i) \cdot \cos\left(\frac{\pi k}{N}(i - 0,5)\right), \qquad k = 0, 1, \cdots, M \tag{2}$$

Diskrétní kosinová transformace (DCT), viz (2), se zde používá místo inverzní diskrétní Fourierovy transformace (IDFT). Dle [2] se vychází z výkonového spektra, které je reálné a symetrické a je tudíž možné IDFT nahradit DCT. Počet MFCC koeficientů se volí podle jejich aplikace. U rozpoznávání řeči je používáno přibližně 13 koeficientů, nicméně při rozpoznávání mluvčího se jich používá přibližně 20. Dále se obvykle používají první a někdy i druhé derivace MFCC koeficientů, které lépe charakterizují dynamiku hlasu. Tyto se označují také jako delta (případně jako delta-delta) koeficienty.



Obr. 2 Banka trojúhelníkových filtrů

2.2.3 Další používané příznaky řeči

Kromě MFCC koeficientů existují i jiné příznaky řeči, které ovšem pro úlohu rozpoznání mluvčího ve spojení s GMM modelem nejsou tolik přesné jako MFCC. V [1] při experimentech identifikace mluvčího autor porovnává úspěšnost modelu např. pro následující příznaky:

- CCEP (koeficienty komplexního kepstra),
- RCEP (koeficienty reálného kepstra),
- LPCC (LPC kepstrální koeficienty),
- MFCC (mel-frekvenční kepstrální koeficienty),
- LFCC (lineární frekvenční kepstrální koeficienty).

U příznaků CCEP a RCEP jsou výpočetní nároky vysoké a častěji se používají LPCC koeficienty, viz dále.

Pro výpočet kepstrálních LPCC koeficientů je použito lineární prediktivní kódování (LPC). Na základě dostatečného množství vzorků signálu lze další vzorek predikovat lineární kombinací vzorků předchozích a vzorků buzení [2]. Přenos modelu řeči je popsán koeficienty (obvykle se značí a_i) a zesílením. Více koeficientů popisuje model přesněji, ale (podobně jako u MFCC) pro více než 20 koeficientů se už přesnost identifikace příliš nezlepšuje [1]. Výhodou této metody je relativně vysoká přesnost a zároveň nízké výpočetní nároky.

Nejlepší úspěšnost identifikace vykazují příznaky MFCC a LPCC. Dle tab. 8.1 [1] byla úspěšnost ve spojení s GMM modelem při identifikaci v uzavřené množině pro 500 mluvčích 99,4 % (MFCC) a 98,8 % (LPCC). U ostatních příznaků jsou výsledky propastně horší.

Koeficienty LFCC jsou počítány podobným způsobem jako MFCC, ale není použita Melovská banka filtrů.

2.3 Hodnocení přesnosti identifikace mluvčího

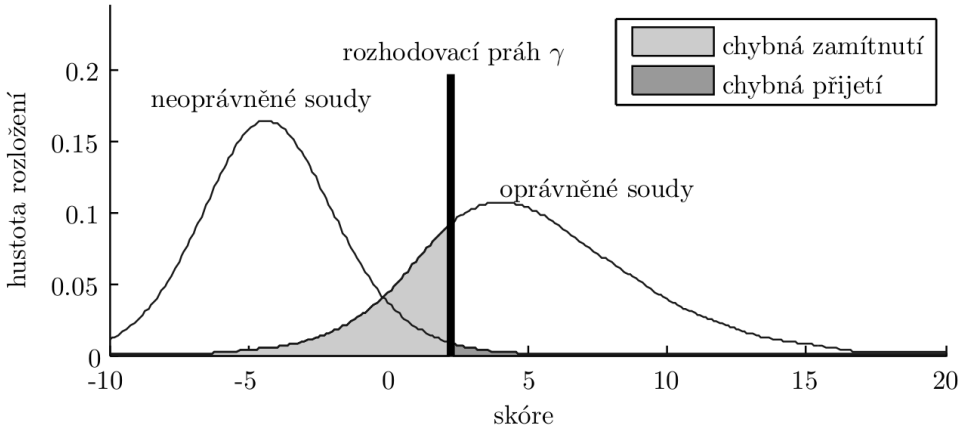
Přesnost identifikace mluvčího se určuje v závislosti na typu systému. Pokud se jedná o systém s uzavřenou množinou mluvčích a rozpoznávaný mluvčí může být pouze z této množiny, vypočte se pouze chyba identifikace. V případě systému s otevřenou množinou mluvčích je situace složitější. Zde je kromě špatné identifikace ještě možné chybně

určit, zda pochází rozpoznávaný (skutečný) mluvčí z množiny referenčních (zapsaných) mluvčích, nebo se jedná o cizího mluvčího [2].

Chybu identifikace mluvčího v uzavřené množině lze vyjádřit vztahem (3),

$$P_{err} = \frac{N_{err}}{N_{total}} \tag{3}$$

kde N_{err} je počet neúspěšných pokusů identifikace a N_{total} je celkový počet pokusů.



Obr. 3 Příklad nastavení verifikačního prahu [3]

U systému v otevřené množině se používá verifikační práh, který označíme γ. Na obr. 3 je typické rozložení pravděpodobnosti hodnoty skóre pro oprávněné a neoprávněné soudy s vyznačeným prahem. Oprávněný soud je takový, kdy správně identifikovaný mluvčí patří do referenční množiny zapsaných mluvčích. Naopak neoprávněný soud zahrnuje cizí mluvčí. Čím vzdálenější jsou od sebe střední hodnoty obou rozložení a čím menší je jejich rozptyl, tím lepší je rozlišovací schopnost verifikace. Cílem je najít vhodný práh pro požadovanou aplikaci.

Za předpokladu správné identifikace mohou ve fázi verifikace nastat čtyři různé situace [3]:

- správné přijetí,
- chybné přijetí (False Acceptance),
- chybné zamítnutí (False Rejection),
- správné zamítnutí.

Míra chybného přijetí v závislosti na stanoveném prahu γ se označuje P_{FA} a je vyjádřena vztahem (4), kde N_{FA} je počet pokusů, kdy došlo k chybnému přijetí mluvčího a $N_{nontarget}$ je počet neoprávněných soudů, které byly pro detekci předloženy.

$$P_{FA}(\gamma) = \frac{N_{FA}(\gamma)}{N_{nontarget}} \tag{4}$$

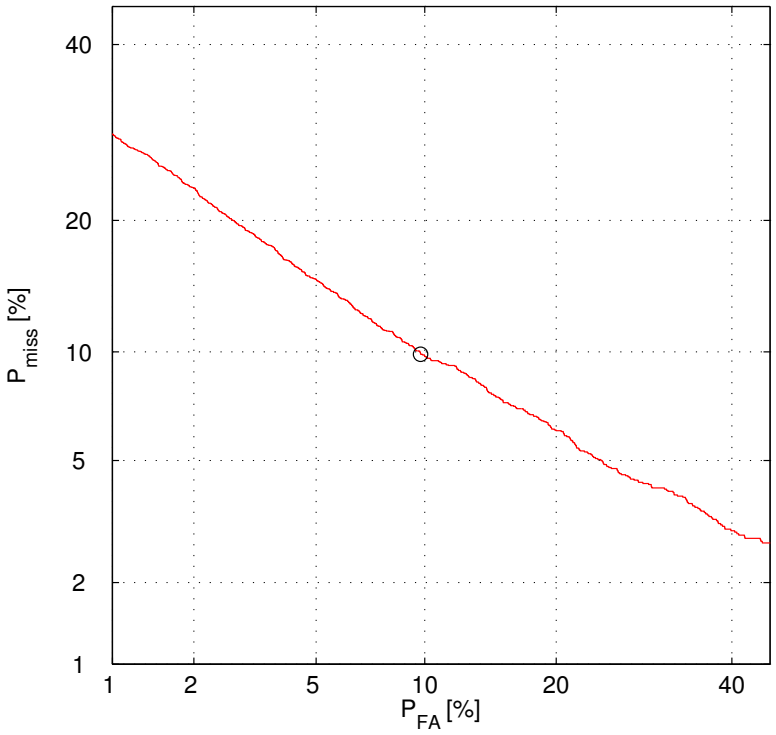
Na druhé straně míru chybného odmítnutí ve vztahu (5) určuje poměr počtu soudů chybných zamítnutí N_{miss} v závislosti na prahu γ a počtu oprávněných soudů N_{target}.

$$P_{miss}(\gamma) = \frac{N_{miss}(\gamma)}{N_{target}} \tag{5}$$

Míra P_{FA} se také někdy označuje jako FAR (False Acceptance Rate) a míra P_{miss} jako FRR (False Rejection Rate). Pro přehledné zobrazení těchto pravděpodobnostních měr

v závislosti na zvoleném prahu se používá DET (Detection Error Trade-off) křivka, viz obr. 4. Rozlišovací schopnost systému lze ohodnotit mírou EER, která je definována vztahem

$$EER = P_{miss}(\gamma) = P_{FA}(\gamma). \tag{6}$$



Obr. 4 Příklad DET křivky generované nástrojem DETware z přiložených evaluačních dat NIST [4]. Vyznačený bod označuje míru stejné chyby EER.

DET charakteristika vychází ze známější ROC (Receiver Operating Characteristics) charakteristiky. U DET křivky nejsou hodnoty P_{miss} a P_{FA} vynášeny na osy v lineárním měřítku jako u ROC charakteristiky, ale měřítka os odpovídají kvantilové funkci standardního normálního rozložení [3]. DET křivka by měla být přibližně lineární se zápornou směrnicí, a pokud bude rozptyl rozložení oprávněných a neoprávněných soudů stejný, bude mít směrnice hodnotu -1. Křivka je tvořena spojnicí diskrétních bodů chybových měr, nicméně výhodou charakteristiky je vyšší rozlišení v oblastech s nižšími chybami obou měr. DET charakteristika je tedy vhodná především pro srovnání více detekčních systémů v jednom grafu, ale v omezeném rozsahu pro nastavení pracovního bodu γ.

2.3.1 DCF funkce

Dle [5] chybová míra EER nekoresponduje s potřebami pro reálné aplikace. Nastavení prahu γ u verifikační fáze detektoru je závislé na typu použité aplikace. Například u bezpečnostní aplikace bude žádoucí zajistit co nejmenší chybu P_{FA}, ale nebude problém připustit větší chybu P_{miss}. Pracovní bod na DET křivce lze vychýlit jedním nebo druhým směrem od pracovního bodu EER. Pro hledání nového pracovního bodu byla

standardizačním úřadem NIST definována metrika DCF (Detection Cost Function) [4], která je definována dle vztahu

$$C_{det}(P_{miss}, P_{FA}) = C_{miss}P_{miss}P_{target} + C_{FA}P_{FA}(1 - P_{target}). \tag{7}$$

K dispozici jsou tři statické parametry, kterými lze pracovní bod vychýlit:

1. C_{miss},
2. C_{FA},
3. P_{target}.

Parametrem C_{miss} je cena za chybné zamítnutí oprávněného soudu. Analogicky C_{FA} je cena za chybné přijetí neoprávněného soudu a poslední parametr P_{target} označuje apriorní pravděpodobnost výskytu oprávněného soudu. Například vyšší hodnota ceny C_{FA} zvyšuje důležitost chybové míry P_{FA}.

Optimální je stanovit rozhodovací práh tak, aby byla hodnota C_{det} minimální na základě vzorce (8) [3]. Hodnota C_{det}^{min} vyjadřuje podobně jako ukazatel EER rozlišovací schopnost verifikačního systému, ale na základě specifických požadavků pro danou aplikaci.

$$C_{det}^{min} = \min_{-\infty \leq \gamma \leq \infty} C_{miss}P_{miss}(\gamma)P_{target} + C_{FA}P_{FA}(\gamma)(1 - P_{target}) \tag{8}$$

Hodnota P_{target} přímo neodpovídá skutečnému podílu oprávněných soudů. Tomuto podílu odpovídá efektivní apriorní pravděpodobnost určená dle vzorce

$$\widetilde{P}_{target} = \frac{P_{target}C_{miss}}{P_{target}C_{miss} + (1 - P_{target})C_{FA}}. \tag{9}$$

Například pro zvolené parametry evaluací NIST do roku 2008 viz [4] a [3] odpovídá $\widetilde{P}_{target}$ hodnotě podílu přibližně 0,092, kde množina neoprávněných soudů $N_{nontarget}$ čítá 30 312 mluvčích [4].

3 Moderní metody rozpoznávání mluvčího

Mezi starší a méně přesné metody (dle experimentů [1]), které používají vzorovou reprezentaci hlasu [2], patří například rozpoznávání založené na časových funkcích nebo na využití vektorové kvantizace. U první zmíněné metody se počítá kumulovaná vzdálenost mezi dvěma promluvami na základě časových funkcí příznakových vektorů (průběh energie, základní frekvence hlasu...). Čím jsou si promluvy podobnější, tím je hodnota vzdálenosti menší. U vektorové kvantizace je reprezentace mluvčího založena na množině centroidů; měřením průměrné vzdálenosti testovaných příznakových vektorů od těchto centroidů je určena míra podobnosti.

Současné moderní metody rozpoznávání mluvčího jsou založené na statistickém modelování. V současné době jsou v této oblasti nejpoužívanější metody založené na směsi Gaussovských hustotních funkcí. Do popředí zájmu se také stále více dostává rozpoznávání s využitím hlubokých neuronových sítí.

3.1 Klasifikace mluvčích na bázi GMM

Klasifikace na bázi GMM, neboli směsí Gaussovských hustotních funkcí, je založena na spojitém rozdělení pravděpodobnosti příznaků řeči, které co nejlépe modelují mluvčího nebo model pozadí. Cílem GMM klasifikace je najít v databázi takový model mluvčího, který bude svou hustotou pravděpodobnosti nejlépe odpovídat testovanému rozložení příznaků řeči přes všechny dimenze příznakového vektoru. Pokud budeme uvažovat například příznaky řeči s 20 MFCC koeficienty, bude potřeba natrénovat GMM model přes 20 dimenzí. V následujících podkapitolách bude nastíněna problematika GMM a její využití pro univerzální model pozadí (Universal Background Model, UBM). Poté budou představeny dva typy systémů, které jsou na UBM založeny.

3.1.1 Matematické pozadí Gaussovských hustotních funkcí

Máme Gaussovské F-dimenzionální rozložení pravděpodobnosti, které je definováno jako $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, kde $\boldsymbol{\mu}$ je vektor střední hodnoty a $\boldsymbol{\Sigma}$ označuje kovarianční matici. Kovarianční matice může být buď diagonální, nebo plná. Každá hustotní funkce je vážena parametrem w_c, kde c je index komponenty modelu.

Pozorujeme-li F-dimenzionální příznakový vektor $\boldsymbol{o}$, budeme pro klasifikační model potřebovat F-dimenzionální střední hodnotu a rozptyl (respektive kovarianční matici), tedy vektor $\boldsymbol{\mu}$ a matici $\boldsymbol{\Sigma}$. Věrohodnostní funkce, tedy pravděpodobnost shody pozorovaného příznakového vektoru s modelem $\boldsymbol{\theta}$, je definována vzorcem (10).

$$P(\boldsymbol{o}|\boldsymbol{\theta}) = \sum_{c=1}^{C} w_c \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{10}$$

Hodnota C udává celkový počet složek (komponent) GMM směsi. Odvození $\mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ je uvedeno v [2] a například v disertační práci [3]. Odvození vychází z vícerozměrného Gaussovského rozložení pravděpodobnosti. Pro parametry modelu w_c jsou stanoveny

podmínky, viz (11). Parametry modelu mluvčího nebo modelu pozadí lze popsat zápisem dle vztahu (12).

$$\sum_{c=1}^{C} w_c = 1 \qquad w_c \geq 0 \tag{11}$$

$$\boldsymbol{\theta} = \{w_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}, \qquad c = 1, \ldots, C \tag{12}$$

Nahrávka rozpoznávaného mluvčího bude obsahovat více příznakových vektorů, tedy posloupnost $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_N\}$. Logaritmus pravděpodobnosti, že nahrávka rozpoznávaného mluvčího odpovídá jeho modelu $\boldsymbol{\theta}$, je vyjádřen vztahem (13) (odvození viz [3]), kde se sčítají logaritmy věrohodnostních funkcí přes všechny příznakové vektory, a N je počet příznakových vektorů. Vztah (13) platí za předpokladu, že jsou příznakové vektory nezávislé. Výpočet konstanty K_c, která je použitá ve vzorci (13), je uveden vztahem (14). F zde značí počet dimenzí příznakového vektoru a $|\boldsymbol{\Sigma}_c|$ je determinant matice $\boldsymbol{\Sigma}_c$.

$$\log P(\boldsymbol{O}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \sum_{c=1}^{C} \exp\left(K_c - \frac{1}{2}\boldsymbol{o}_n^T \Sigma_c^{-1} \boldsymbol{o}_n + \boldsymbol{o}_n^T \boldsymbol{\Sigma}_c^{-1} \boldsymbol{\mu}_c\right) \tag{13}$$

$$K_c = \log w_c - \frac{1}{2} F \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}_c| - \frac{1}{2} \boldsymbol{\mu}^T \boldsymbol{\Sigma}_c^{-1} \boldsymbol{\mu} \tag{14}$$

Aby bylo možné rozpoznávat promluvy jednotlivých mluvčích, je potřeba vytvořit modely mluvčích. Postupy vytvoření modelů mluvčích budou popsány dále. Obecně se pro hledání parametrů modelu, při kterém chceme dosáhnout maximální věrohodnosti, používá iterační algoritmus EM (Expectation Maximization). Algoritmus se skládá ze dvou opakujících se kroků [3]:

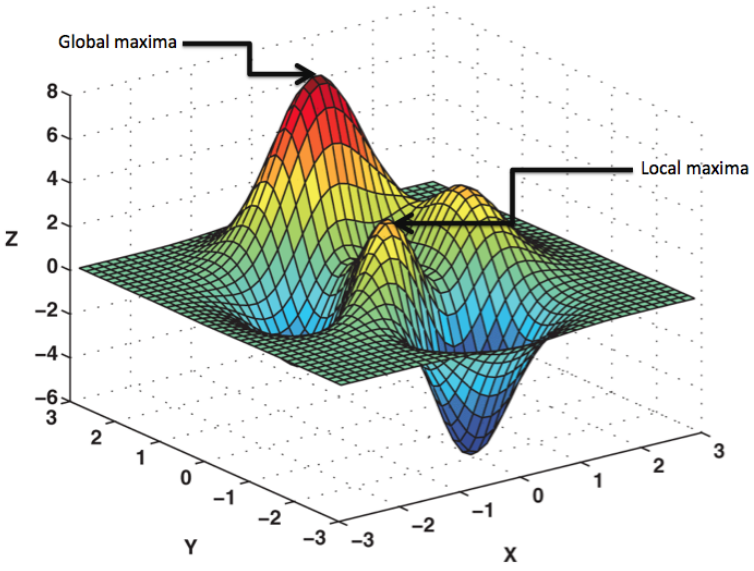
- Expectation – vyhodnocení věrohodnosti z právě odhadnutých parametrů.
- Maximization – hledání takových parametrů, které maximalizují věrohodnost.

Opakováním těchto dvou kroků je nalezeno maximum a algoritmus je ukončen. Jedná se o výpočetně náročnou operaci, u které není navíc jisté, zda bude nalezeno globální maximum. Algoritmus se může zastavit i na lokálním maximu, jak je znázorněno na obr. 5 od autora webové publikace [6], ve které je algoritmus podrobně vysvětlen. Obrázek znázorňuje GMM směs pro příznakový vektor dimenze $F = 2$. Inicializace parametrů je dle [2] nejvýhodnější, když jsou parametry modelu dle zápisu (12) před spuštěním EM algoritmu zcela náhodné.

3.1.2 Univerzální model pozadí – UBM

Trénovat statistický model GMM pro každého jednotlivého mluvčího by bylo náročné jak na výpočetní výkon, tak především na paměť. Je tedy vhodné natrénovat takový univerzální model, který bude reprezentovat možné rozložení příznaků řeči skupiny mluvčích, a jednotlivé mluvčí z tohoto modelu odvodit neboli adaptovat. UBM je trénován z velkého množství mluvčích a je na mluvčím nezávislý.

Při trénování UBM modelu je třeba vybrat trénovací data tak, aby byla ve stejné doméně, v jaké bude model použit pro rozpoznávání. Doménou je myšlena stejná vlastnost obou skupin nahrávek (trénovacích a rozpoznávaných), například použití typově stejných mikrofonů, nebo přenosových kanálů, jež do akustického signálu zavádí konvoluční šum.



Obr. 5 Hledání globálního maxima ve směsi hustotních funkcí (příznakový vektor o dvou dimenzích) [6].

Doménou může být myšleno také pohlaví mluvčích. Pokud bude v úloze rozpoznávání model použit pro obě pohlaví rovnoměrně, je důležité, aby také při trénování byly použity nahrávky rovnoměrně rozdělené mezi muže a ženy. V opačném případě by se model vychyloval k jedné nebo druhé skupině. Pokud je k dispozici například více trénovacích dat pro muže, je možné dle [2] natrénovat dva modely odděleně. V tom případě bude jeden model natrénován pro muže a druhý pro ženy. Následně je lze spojit oba v jeden model.

Důležitým parametrem pro trénovaní UBM, který musí být před započetím dané úlohy pečlivě zvolen, je počet GMM komponent. Obecně platí, že čím více bude k dispozici dat pro natrénování UBM modelu, tím více GMM komponent bude potřeba pro jeho popis. Pokud bude například k dispozici několik hodin nahrávek promluv, vystačíme si se 128 komponentami, ale pro desítky až stovky hodin bude nutné zvolit 1 024 nebo 2 048 komponent [3]. Počet komponent se obvykle volí jako mocnina dvou, jedná se ale o zvyk, nikoliv o nutnost. Při volbě komponent u většího množství dat je zpravidla limitující výpočetní výkon a také paměťové nároky trénující aplikace.

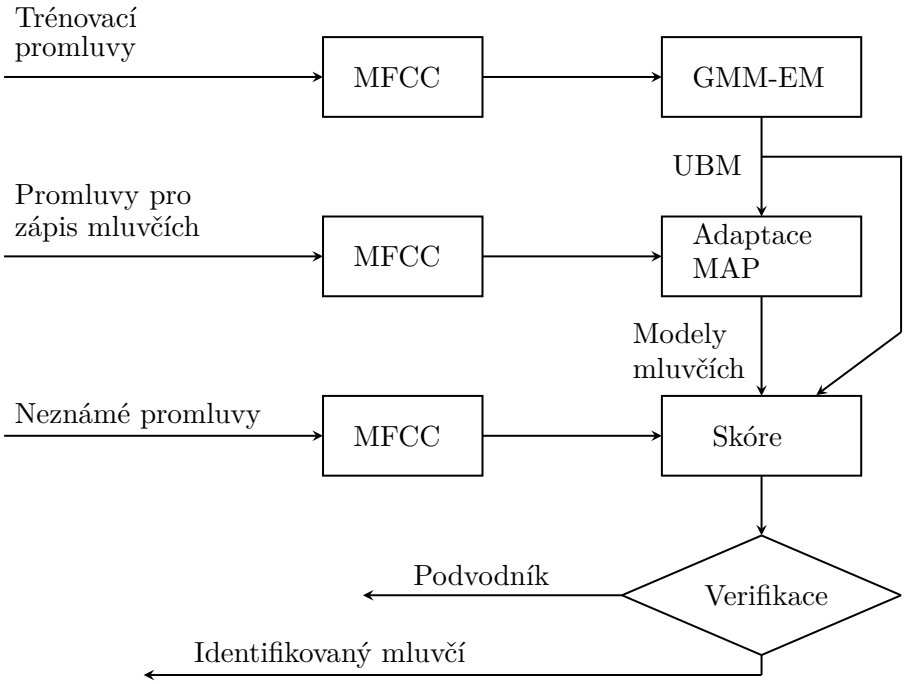
Problém také nastává, pokud bude pro UBM zvoleno příliš mnoho GMM komponent vzhledem k množství nahrávek. Dle [3] lze model přetrénovat, k nepřesnostem dochází na okrajích příznakového prostoru během hledání parametrů modelu pomocí EM algoritmu. Pro okrajové komponenty nastává situace, ve které mají přiřazen pouze jeden z příznakových vektorů, taková komponenta bude mít nulovou kovarianční matici $\mathbf{\Sigma}_c$ a Gaussovská křivka poté bude zredukována do jednoho bodu. Existují mechanizmy, které tento problém alespoň částečně řeší, nicméně se jedná o výrazný jev.

3.1.3 Systém UBM-GMM

Univerzální model pozadí se používá v různých uskupeních s dalšími navazujícími výpočty. Možné řešení je v literatuře často nazýváno jako systém UBM-GMM. U trénování UBM z příznaků řeči množiny mluvčích, na kterých bude prováděna identifikace, se používá metoda maximálně věrohodného odhadu (ML) parametrů GMM. Pro odvozování modelů mluvčích z UBM modelu se používá metoda maximální aposteriorní pravděpodobnosti (MAP), viz [3]. Modely mluvčích jsou z UBM odvozeny adaptací pouze vektorů středních hodnot $\boldsymbol{\mu}_c$, viz parametry GMM modelu (12). U metody odhadu maximální věrohodnosti (ML) hledáme statické parametry, které nejsou předem známy, zatímco odhad metodou MAP je založen na hledání parametrů, jež jsou náhodnými veličinami, a předpokládá se, že rozložení jejich pravděpodobnosti je známé.

Předmětem systému UBM-GMM jsou následující kroky:

- trénování UBM modelu metodou ML,
- odvození modelů mluvčích metodou MAP,
- výpočet skóre pro identifikaci mluvčího.



Obr. 6 UBM-GMM systém, adaptace mluvčích z UBM

Na obr. 6 je naznačen princip systému UBM-GMM [7] použitého u identifikace v otevřené množině, u kterého je trénovací sekvence nahrávek promluv použita ve dvou fázích, poté je ve fázi třetí prováděn výpočet skóre. Za předpokladu, že máme k dispozici sekvenci příznakových vektorů $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_N\}$ od jednoho mluvčího, můžeme skóre vyjádřit vztahem (15), kde $\boldsymbol{\theta}_s$ označuje model mluvčího s a $\boldsymbol{\theta}_{UBM}$ označuje společný model pozadí [3]. Ze vztahu plyne, že čím větší bude věrohodnost $P(\boldsymbol{o}_n|\boldsymbol{\theta}_s)$ proti věrohodnosti pozadí pro co největší počet příznakových vektorů, tím více bude podpořena hypotéza, že rozpoznávaná promluva patří právě mluvčímu s.

$$score_{UBM-GMM} = \frac{1}{N}\sum_{n=1}^{N} \log \frac{P(\boldsymbol{o}_n|\boldsymbol{\theta}_s)}{P(\boldsymbol{o}_n|\boldsymbol{\theta}_{UBM})} \tag{15}$$

3.1.4 Systém reprezentace mluvčích pomocí i-vektorů

Konkrétního mluvčího v systému UBM-GMM charakterizuje takzvaný supervektor. Jedná se o zřetězený vektor středních hodnot $\boldsymbol{\mu}_c$ přes všechny komponenty modelu, který byl adaptován z UBM modelu. Dimenze supervektoru je $C \cdot F$, kde C je počet komponent GMM modelu a F dimenze příznakového vektoru [3]. Jelikož mají supervektory mnoho nadbytečné informace, hledají se takové vektory (i-vektory), které budou stále charakterizovat konkrétního mluvčího, ale budou mít menší dimenzi. I-vektor není na rozdíl od supervektoru přímo pozorovaný vektor.

Vznik názvu i-vektor může být chápan například od slova „střední“ (intermediate), kde střední je velikost dimenze [3]. Dimenze i-vektoru je vyšší než dimenze příznakových vektorů F a nižší než dimenze supervektorů. Princip odvození i-vektorů je založen na statistickém nástroji, na takzvané faktorové analýze.

„Cílem faktorové analýzy je provést vyjádření variability pozorovatelných proměnných na základě menšího počtu skrytých (latentních) proměnných, tzv. faktorů.“ [3]

Uvažujme zjednodušený model faktorové analýzy dle vztahu (16), kde máme k dispozici supervektor středních hodnot $\boldsymbol{\mu}$ (střední hodnoty UBM modelu), který bude nezávislý na mluvčím a na konkrétní nahrávce. Index r označuje konkrétní nahrávku a index s mluvčího. Supervektor $\boldsymbol{m}_{r,s}$ odpovídá jedné konkrétní nahrávce od daného mluvčího. Matice $\boldsymbol{T}$ použitá pro redukci dimenze supervektorů se někdy označuje také jako hyper-parametry a její dimenze je $C \cdot F \times D_{ivec}$, kde D_{ivec} je dimenze i-vektoru $\boldsymbol{x}_{r,s}$. I-vektor má rozložení pravděpodobnosti $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, kde $\boldsymbol{I}$ je jednotková kovarianční matice a střední hodnoty vektoru jsou nulové.

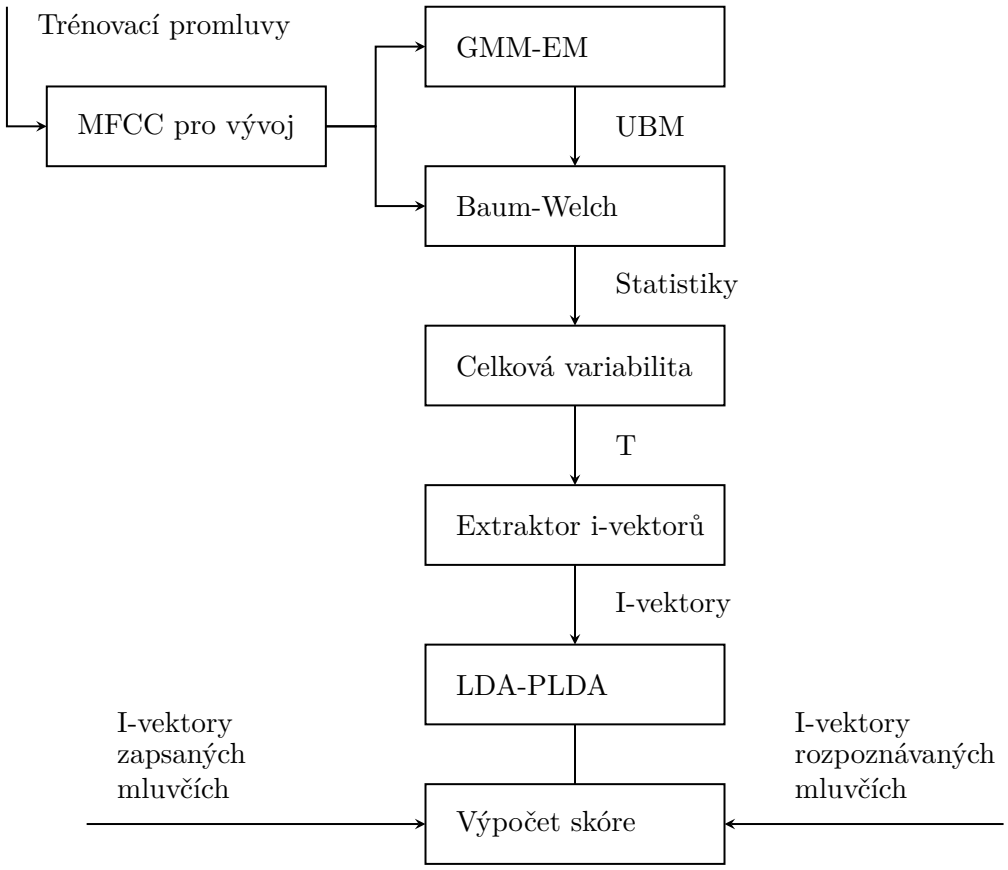
$$\boldsymbol{m}_{r,s} = \boldsymbol{\mu} + \boldsymbol{T}\boldsymbol{x}_{r,s} \tag{16}$$

Princip trénování a použití modelu dle [7] je naznačen na obr. 7. Podobně jako u systému UBM-GMM je nejprve natrénován UBM model. Dále jsou Baum-Welch algoritmem (dle [2] se jedná o speciální případ EM algoritmu) napočítány postačující statistiky pomocí okupační pravděpodobnosti a jsou centrovány kolem vektoru $\boldsymbol{\mu}$ [3]. Dále je vytvořen prostor celkové variability, který definuje matice $\boldsymbol{T}$. Následně je možné provést extrakci i-vektorů pro konkrétní nahrávky jednotlivých mluvčích dle vztahu (16). I-vektory trénovacích promluv je možné dále použít k natrénovaní PLDA modelu nebo pomocí LDA snížit velikost dimenze D_{ivec} a následně vyhodnotit skóre. Další možností je obě metody nakombinovat v pořadí LDA a PLDA [8]. Obě metody budou vysvětleny v samostatné podkapitole.

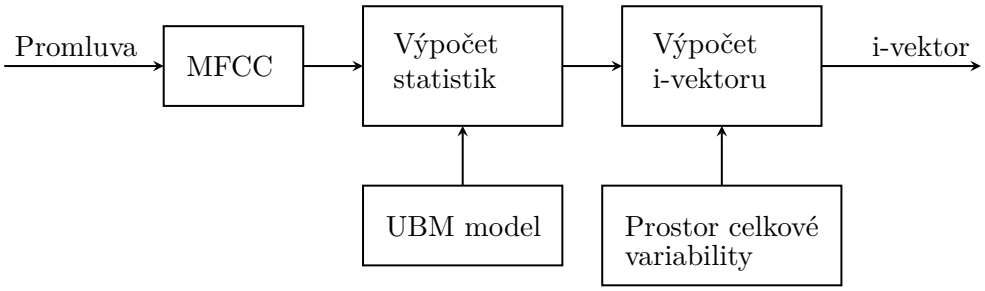
U modelu na obr. 7 je skóre vyhodnoceno až po natrénování PLDA modelu. Pro vytvoření PLDA modelu a případně LDA transformační matice jsou použity i-vektory z trénovacích promluv a pro zápis mluvčích je použita jiná množina nahrávek. Je ale také možné skóre vyhodnotit přímo mezi i-vektory trénovacích a rozpoznávaných nahrávek ihned po extrakci i-vektorů v trénovací větvi a bez použití PLDA. Zapisovaní mluvčí (respektive jejich i-vektory) jsou tak zároveň použiti pro natrénování modelu.

Výpočet i-vektorů z nahrávek mluvčích je prováděn dle obr. 8 (viz [3] a [9]). Tento proces lze považovat i za parametrizaci promluvy. Před vyhodnocením skóre (mezi i-vektory mluvčích) se obvykle vypočte průměr z i-vektorů jednotlivých nahrávek konkrétního mluvčího.

Pro vyhodnocení míry podobnosti (skóre) mezi dvěma i-vektory, na jedné straně i-vektoru zapsaného mluvčího a na druhé straně i-vektoru rozpoznávaného mluvčího, se používá kosinová vzdálenost dle vztahu (17). V čitateli zlomku se vypočte skalární



Obr. 7 Princip modelu založeného na i-vektorech.



Obr. 8 Postup vytvoření i-vektoru pomocí již natrénovaného modelu

součin vektorů a ve jmenovateli se násobí délky obou i-vektorů. Čím vyšší je skóre, tím více je podpořena hypotéza, že i-vektory $\boldsymbol{x}_1$ a $\boldsymbol{x}_2$, které jsou odvozeny z různých nahrávek, pochází od stejného mluvčího.

$$score_{ivec}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle}{\|\boldsymbol{x}_1\| \cdot \|\boldsymbol{x}_2\|} \tag{17}$$

3.2 Metody pro zvýšení přesnosti klasifikace

3.2.1 Lineární diskriminační analýza – LDA

Cílem LDA (Linear Discriminant Analysis) v úloze rozpoznávání mluvčího je za pomoci lineární transformace i-vektorů zvýšit schopnost rozlišovat během klasifikace mezi jednotlivými mluvčími, vedlejším efektem je snížení dimenze i-vektorů. Dle [3] je LDA lineární transformace

$$\boldsymbol{x}' = \boldsymbol{A}^T \boldsymbol{x}, \tag{18}$$

kde $\boldsymbol{x}$ je původní n-dimenzionální i-vektor a $\boldsymbol{x}'$ je nový i-vektor se sníženou dimenzí m. Transformační matice má tedy rozměr $n \times m$.

Pro lepší rozlišitelnost mezi i-vektory mluvčích (obecně mezi třídami)[2] je pomocí transformace zajištěn větší rozptyl mezi třídami a menší rozptyl uvnitř tříd. Rozlišením uvnitř tříd se myslí situace, kdy je v trénovací množině k dispozici více promluv (i-vektorů) od jednoho mluvčího.

Optimalizaci v závislosti na transformační matici A je možné provést dle vztahu

$$J(\boldsymbol{A}) = \mathrm{tr}\left((\boldsymbol{A}^T \boldsymbol{\Sigma}_W \boldsymbol{A})^{-1} (\boldsymbol{A}^T \boldsymbol{\Sigma}_B \boldsymbol{A})\right) \tag{19}$$

maximalizací kritéria $J(\boldsymbol{A})$[1].

„Maximalizace tohoto kritéria bude dosažena, pokud budou sloupce transformační matice $\boldsymbol{A}$ *tvořeny vlastními vektory příslušnými* m *největším vlastním číslům matice* $\boldsymbol{\Sigma}_W^{-1}\boldsymbol{\Sigma}_B$*.“* [3]

Matice $\boldsymbol{\Sigma}_W$ je celková kovariance uvnitř tříd počítaná přes všechny třídy a $\boldsymbol{\Sigma}_B$ je kovariance mezi třídami. Obě kovarianční matice jsou nadefinovány z trénovacích dat v prostoru dimenze n. Kovariance $\boldsymbol{\Sigma}_W$ je spočítána jako

$$\boldsymbol{\Sigma}_W = \sum_{s=1}^{S} \boldsymbol{\Sigma}_s, \tag{20}$$

přes kovariance jednotlivých mluvčích s, kde S je celkový počet mluvčích. Ve vzorci (20) je $\boldsymbol{\Sigma}_s$ napočítána z i-vektorů, které reprezentují promluvy konkrétního mluvčího s. Tato kovariance je vypočtena ze vztahu

$$\boldsymbol{\Sigma}_s = \sum_{r=1}^{R_s} (\boldsymbol{x}_r - \boldsymbol{\mu}_s)(\boldsymbol{x}_r - \boldsymbol{\mu}_s)^T, \tag{21}$$

kde R_s je počet nahrávek od daného mluvčího a r označuje konkrétní nahrávku (respektive její i-vektor). Vektor $\boldsymbol{\mu}_s$ je vypočten jako odhad středních hodnot i-vektorů konkrétního mluvčího z jeho promluv.

Kovariance mezi třídami $\boldsymbol{\Sigma}_B$ je definována jako

$$\boldsymbol{\Sigma}_B = \sum_{s=1}^{S} R_s(\boldsymbol{\mu}_s - \boldsymbol{\mu}_T)(\boldsymbol{\mu}_s - \boldsymbol{\mu}_T)^T, \tag{22}$$

kde $\boldsymbol{\mu}_T$ je odhad středních hodnot přes i-vektory všech nahrávek od všech mluvčích.

Omezením při aplikaci LDA transformace je maximální velikost nové dimenze m. Pro m-dimenzionální výstup je nutné mít v trénovací množině k dispozici $(m+1)$ mluvčích [3]. Při požadavku na velikost dimenze například v řádu stovek a při použití menší množiny mluvčích u trénovacích dat není LDA možné použít.

[1] Operátor $tr(\cdot)$ provede součet prvků matice na diagonále.

3.2.2 Pravděpodobnostní lineární diskriminační analýza – PLDA

PLDA byla navržena jako pravděpodobnostní alternativa (viz [10]) k lineární diskriminační analýze (LDA). Cílem je tedy opět zlepšit rozlišitelnost mezi jednotlivými mluvčími v prostoru i-vektorů; zde ale nedochází ke snižování dimenze jako u LDA. PLDA vychází z faktorové analýzy [3] a je pro ni definován model

$$\boldsymbol{x}_{r,s} = \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{y}_s + \boldsymbol{U}\boldsymbol{w}_{r,s} + \boldsymbol{\epsilon}_{r,s}, \tag{23}$$

$$\boldsymbol{s}_s = \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{y}_s, \tag{24}$$

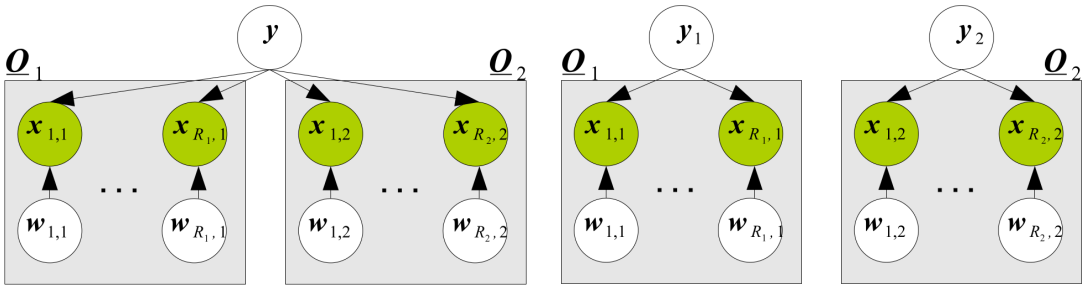
$$\boldsymbol{c}_{r,s} = \boldsymbol{U}\boldsymbol{w}_{r,s} + \boldsymbol{\epsilon}_{r,s}, \tag{25}$$

kde $\boldsymbol{x}_{r,s}$ je i-vektor vstupní nahrávky s indexem r mluvčího s. Složka modelu $\boldsymbol{s}_s$ vyjadřuje charakteristiku mluvčího s a je stejná pro všechny vlastní nahrávky. Druhá složka $\boldsymbol{c}_{r,s}$ vyjadřuje proměnné akustické podmínky prostředí a analogové části nahrávacího zařízení. Dále $\boldsymbol{\epsilon}_{r,s}$ vyjadřuje variabilitu prostředí, kterou se nepodařilo definovat ostatními členy modelu s rozložením $P(\boldsymbol{\epsilon}_{r,s}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, kde $\boldsymbol{\Sigma}$ je diagonální kovarianční matice. Rozložení pravděpodobnosti proměnných (faktorů) $\boldsymbol{y}_s$ a $\boldsymbol{w}_{r,s}$ je normální, a to $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

Parametry PLDA modelu, které je potřeba natrénovat, jsou:

- zátěžová matice $\boldsymbol{V}$ (model rozdílů mezi mluvčími),
- vektor $\boldsymbol{\mu}$,
- zátěžová matice $\boldsymbol{U}$ (model akustických podmínek),
- matice $\boldsymbol{\Sigma}$.

Trénování parametrů modelu je uskutečněno pomocí EM algoritmu, kdy je zaručena stále rostoucí věrohodnost při vyhodnocování. Při trénování PLDA modelu je potřeba mít k dispozici více nahrávek od každého mluvčího. Čím větší je variabilita akustického prostředí a čím více nahrávek pro daného mluvčího je k dispozici, tím efektivnější jsou výsledky při rozpoznávání oproti základním metodám (např. pouze kosinová vzdálenost).



a) Totožný mluvčí $\boldsymbol{y}$ (hypotéza $\mathcal{H}_s$). **b)** Odlišný mluvčí $\boldsymbol{y}_1$ a $\boldsymbol{y}_2$ (hypotéza $\mathcal{H}_d$).

Obr. 9 Hypotézy, zda mají dvě skupiny nahrávek $\boldsymbol{x}_{r,s}$ trénovacích a rozpoznávaných dat společného mluvčího (9a) nebo nikoli (9b) [3].

Na obr. 9 jsou dvě hypotézy situací, které mohou nastat při rozpoznávání. U každé situace jsou dvě sady nahrávek: $\boldsymbol{O}_1$ a $\boldsymbol{O}_2$. V každé sadě jsou všechny nahrávky vždy od jednoho mluvčího. Dle [10] jsou při porovnání dvou i-vektorů domnělého mluvčího, které reprezentuje odhad střední hodnoty i-vektorů jejich nahrávek v sadě, testovány tyto dvě hypotézy označené jako $\mathcal{H}_s$ a $\mathcal{H}_d$. Cílem je rozhodnout, zda obě sady nahrávek

pocházejí od stejného mluvčího, vyjádřením této podobnosti je skóre dle vztahu (26). Skóre je vyjádřeno jako logaritmus poměru věrohodností testovaných hypotéz.

$$score_{PLDA} = \log \frac{P(\boldsymbol{O}_1, \boldsymbol{O}_2|\mathcal{H}_s)}{P(\boldsymbol{O}_1, \boldsymbol{O}_2|\mathcal{H}_d)}, \tag{26}$$

3.3 Klasifikace mluvčích na bázi DNN

V této práci jsem se zaměřil především na klasifikaci mluvčích na bázi GMM a jejich reprezentaci pomocí i-vektorů. Teoretický popis a implementace klasifikace na bázi DNN (Deep Neural Network) přesahuje rámec této práce.

Některé metody, jako například LDA a PLDA, se v případě DNN používají stejným způsobem jako u GMM za předpokladu, že je u DNN modelu zachována reprezentace pomocí i-vektorů. Dle [11] a [9] se jako příznaky u DNN používají také MFCC koeficienty, nicméně ty jsou používány k natrénování neuronové sítě pro rozpoznávání řeči ASR (Automatic Speech Recognition). Výstup DNN sítě může být dále použit ve formě příznaků BNF (Bottleneck Features) [12] k natrénování UBM/i-vektor modelu, který je v této práci implementován a otestován.

U tohoto modelu (BNF/GMM) je dle studie [11] u chybové míry EER dosahováno až o 26 % lepších výsledků oproti výchozímu modelu MFCC/GMM. Pro trénování a testování modelu byla použita data z evaluací NIST z let 2004–2008 a v obou případech jsou mluvčí reprezentováni pomocí i-vektorů. Nicméně je nutno podotknout, že trénování hluboké neuronové sítě je v porovnání s výchozím modelem výpočetně velice náročná záležitost. Trénování DNN vyžaduje také velké množství dat. Pro aplikace dimenzované na relativně malý počet mluvčích se u současných výpočetních technologií hodí spíše MFCC/GMM model (viz 3.1.4).

Softwarová simulace neuronové sítě na klasických výpočetních systémech je velmi neefektivní. Využití umělých neuronových sítí má nicméně slibnou budoucnost při použití v nízkoúrovňovém modelu, například s memristory. Memristor je vlastně rezistor s pamětí a jeho okamžitá hodnota odporu je dána předchozí hodnotou přiloženého napětí. Memristor může simulovat synapse neuronů, poté by neuronová síť mohla být energeticky výrazně efektivnější. Implementace malé neuronové sítě na bázi memristorů byla realizována v roce 2015 (viz [13]).

4 Implementace

V minulé kapitole popsané nebo zmíněné metody identifikace mluvčího založené na GMM a DNN není nutné implementovat kompletně celé. Lze použít i nástroje, jež se statistickými modely a neuronovými sítěmi pracují.

Jedním z těchto nástrojů v oblasti rozpoznávání mluvčího je nástroj HTK (Hidden Markov Model Toolkit)[14]. Počátky vývoje jsou datovány kolem roku 1993. Dokumentace je na vysoké úrovni (HTK book) a nástroj je takzvaně „user-friendly“.

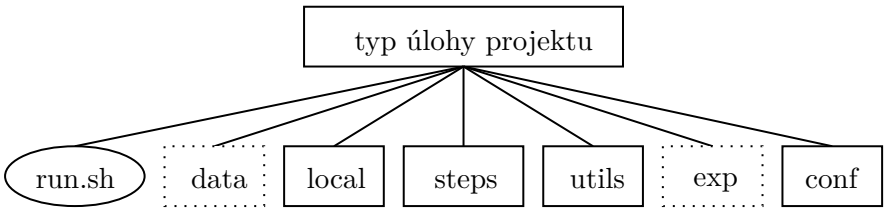
Dobrou mladší alternativou jsou nástroje KALDI (The Kaldi Speech Recognition Toolkit) [15]. Počátky tohoto projektu jsou oficiálně datovány od roku 2009 (Johns Hopkins University). V roce 2010 na projektu participovali akademici z Vysokého učení technického v Brně. Rozsáhlá knihovna KALDI je dynamicky vyvíjena a její zdrojový kód je psán tak, aby byl snadno modifikovatelný a rozšiřitelný. Součástí nástrojů jsou i programy ke konverzi stávajících modelů v HTK do KALDI.

4.1 KALDI nástroje – úvod

Pro prezentovanou implementaci rozpoznávání mluvčího byla zvolena sada nástrojů KALDI. Jde o moderní a multiplatformní řešení, které je dobře otestováno a odladěno na operačním systému GNU/Linux. Jedná se o projekt vyvíjený v modelu OSS (Open-Source Software) [9] a zdrojové kódy a skripty jsou programovány pod licencí Apache verze 2.0. Nástroje je tedy možné díky licenční politice a efektivní implementaci použít pro reálné aplikace.

Velkou výhodou sady nástrojů KALDI je dostupnost ukázkových skriptů, které demonstrují typické řešení různých úloh. Nástroje jsou vyvíjeny především pro účely rozpoznávání řeči, ale jsou k dispozici i ukázkové skripty použité pro rozpoznávání mluvčího na evaluacích NIST v letech 2008 a 2010. V roce 2008 bylo implementováno rozpoznávání mluvčího pomocí GMM směsí a v roce 2010 byly pro srovnání používány i systémy s neuronovými sítěmi DNN převzaté včetně transkripce z úloh rozpoznávání řeči.

Zdrojový kód matematicky výpočetně náročných programů je napsán v objektovém jazyce C++. Výpočetně nenáročné rutiny (například inicializační) jsou psány ve skriptovacím jazyce Perl nebo v terminálovém interpretovaném jazyce BASH. Jazyk Perl je díky regulárním výrazům vhodný zejména pro zpracování transkripce a obecně pro přípravu pomocných textových souborů, díky kterým mohou jednotlivé nástroje následně vyhledat a zpracovat například binárně uložené nahrávky.



Obr. 10 Adresářová struktura KALDI dle zavedené konvence

Na obr. 10 je znázorněna obecná konvence adresářové struktury projektů (ukázkové skripty), kterým se v KALDI říká „recepty“ (recipes) a mnoho jich je dostupných jako součást distribuce. Jako hlavní spustitelný soubor se v každém „receptu“ používá `run.sh`. Zde bývá implementován postup řešení dané úlohy a odtud jsou volány dílčí procedury. Tyto procedury vykonávají jednotlivé výpočetní kroky, jako například přípravu dat nebo výpočet příznaků řeči.

Programy v podobě binárních spustitelných souborů jsou spouštěny převážně v rámci dílčích skriptů (procedur). Jejich chybový výstup (stderr) je obvykle směrován do log souborů v příslušných složkách experimentů, kde jsou k dispozici pro případné ladění nebo hledání chyb.

Adresáře `data` a `exp` označené na obr. 10 tečkovaně jsou vždy vytvořeny v průběhu spuštění hlavního skriptu. Adresář `data` se plní seznamy indexů pro další zpracování nahrávek během inicializační fáze, jak je popsáno výše. Naproti tomu do adresáře `exp` jsou ukládány vypočtené modely experimentů a dále jejich výsledky a mezivýsledky. V adresáři `conf` bývají statické konfigurace, například pro výpočet MFCC koeficientů. Adresář `utils` nabízí řadu pomocných skriptů pro práci se strukturovanými textovými seznamy dle standardů KALDI. Obsahem adresáře `steps` jsou skripty, jež vykonávají jednotlivé výpočetní kroky, které jsou klíčové pro danou úlohu[1]. Konečně adresář `local` je určen pro dílčí procedury, které jsou specifické pro právě implementovanou úlohu.

4.2 Úloha rozpoznávání mluvčího v KALDI

Řešení úlohy rozpoznávání mluvčího založeného na modelu i-vektorů (viz 3.1.4) se skládá typicky z několika kroků, které reprezentují dílčí fáze (stages), viz následující seznam.

- STAGE 0 – Příprava a formátování dat pro KALDI.
- STAGE 1 – Výpočet MFCC a VAD.
- STAGE 2 – Natrénování UBM modelu a i-vektor extraktoru.
- STAGE 3 – Extrakce i-vektorů.
- STAGE 4 – Výpočet skóre pro zvolené metody a vyhodnocení chyby identifikace.
- STAGE 5 – Verifikace a vyhodnocení EER.

Jednotlivé fáze budou popsány ve stejnojmenných podkapitolách.

Adresářová struktura implementace úlohy v1[2] je naznačena na obr. 11. V dalším textu této kapitoly bude na obrázek postupně odkazováno.

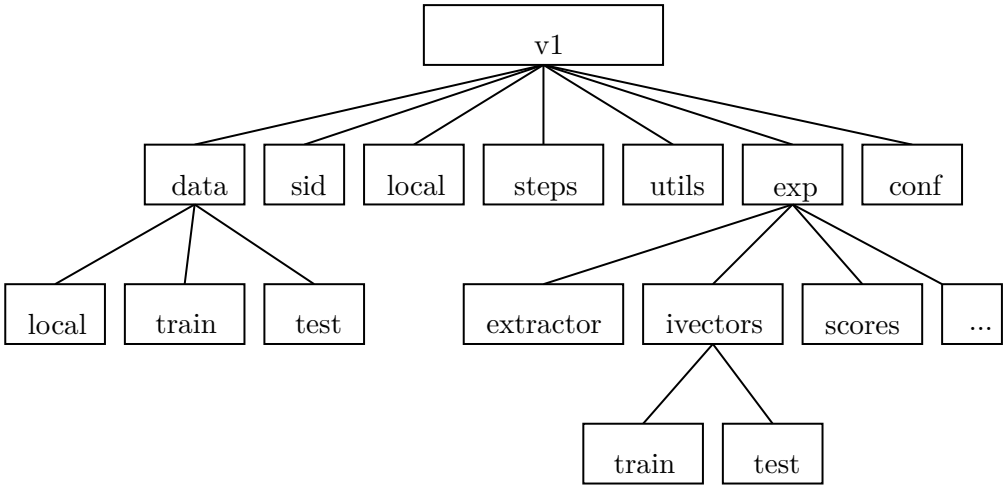
Projekty bývají řešeny modulárně. Pokud řešená úloha vyžaduje práci nad vícero databázemi s podobnými vlastnostmi, jsou jednotlivé databáze nahrávek připraveny do požadovaného formátu v samostatných skriptech. V hlavním skriptu `run.sh` jsou poté výpočty prováděny nad všemi databázemi při jednom průchodu. V prezentované implementaci jsou databáze testovány odděleně právě pro své specifické vlastnosti.

Pro implementaci úlohy rozpoznávání mluvčího jsem zvolil databáze SPEECON a SpeechDat (viz kapitola 5). Každá databáze je implementována do vlastní adresářové struktury a obě implementace jsou si velmi podobné. Rozdíl je v lehce odlišné přípravě dat, zároveň jsou některé výchozí uživatelské proměnné nastaveny jinak. V případě rozdílných parametrů nebo implementace dílčích procedur bude uvedeno, které databáze se popisované vlastnosti týkají.

Hlavní skript `run.sh` je koncipován do dvou částí. V první řadě je zde část pro uživatelské nastavení proměnných, ve které jsou proměnné definovány velkými písmeny

[1] V úlohách rozpoznávání mluvčího je pro tyto účely používán ještě adresář `sid`.

[2] V KALDI se úloha rozpoznávání mluvčího s použitím GMM modelu označuje jako v1.



Obr. 11 Adresářová struktura úlohy v1 (implementace rozpoznávání mluvčího)

a pomocí komentářů v angličtině je stručně vysvětlen jejich význam. Ve druhé řadě je pak programová část (implementuje jednotlivé fáze), ve které jsou pomocné proměnné definovány standardně malými písmeny.

Vstupní uživatelské parametry jsou rozděleny do sekcí podle významu a místa použití, viz následující seznam.

- Způsob spouštění úlohy z hlediska CPU.
- Výběr fáze výpočtu.
- Proměnné pro přípravu dat.
- Úprava vzorkování a kvantování signálu.
- Nastavení parametrů UBM modelu a i-vektor extraktoru.
- Výběr metod pro výpočet skóre a jejich nastavení.

Systém rozpoznávání mluvčího může být spuštěn buď na jednom PC s využitím paralelizace na jednotlivých procesorech, případně může být výpočet distribuován do sítě propojených serverů, v tomto případě pomocí SGE (Sun Grid Engine). Podle přiřazeného pořadového čísla do proměnné `STAGE` je spuštěna pouze jedna vybraná fáze. Zbylé čtyři sekce proměnných jsou popsány u jednotlivých fází.

Jednotlivé fáze lze spouštět samostatně, což je výhodné pro ladění nových výpočtů. Před započetím následující fáze je provedena ruční kontrola předchozího výpočtu a v případě nečekaných problémů (například s daty) není zbytečně plýtván výpočetní výkon na trénování nechtěného modelu. Po odladění vstupních uživatelských parametrů pro konkrétní typ experimentu je možné sérii výpočtů automatizovat.

Pro automatizované provedení všech potřebných fází byly připraveny skripty s prefixem `wrapper`. Každá série experimentů je v některém ohledu specifická. U experimentů, u kterých se průběžně mění například dimenze MFCC, je nutné opakovaně počítat příznaky i celý model. Pokud bude ale měněn parametr počet GMM komponent, příznaky budou připraveny pouze jednou a výpočet modelu i skóre bude proveden vícekrát. Pro každou sérii experimentů byl vytvořen vlastní `wrapper.sh` skript, který volá jádro úlohy v1 (`run.sh`) s požadovanými parametry.

Se složkami `data` a `exp` (viz obr. 11) se z hlediska fází pracuje odděleně. Je možné jednou vytvořit model z trénovacích dat a následně opakovaně zapisovat mluvčí pomocí i-vektorů například s jinou množinu nahrávek. Bude se tedy opakovat fáze pro přípravu dat a výpočet MFCC a následně bude rovnou použita fáze pro extrakci i-vektorů. Bude tedy vynechán výpočetně náročný krok trénování modelu. První dvě fáze jsou vázány

na adresář `data`, kdežto v dalších fázích jsou vytvářeny soubory a podadresáře do složky `exp`.

V posledních dvou fázích (STAGE 4 a 5) je u výpočtu skóre a verifikace možné si zvolit, které metody budou použity (LDA, PLDA...). Nutno podotknout, že u těchto výpočtů není implementováno paralelní zpracování dat, nicméně jejich použití není zdaleka tak náročné na čas procesoru jako u trénování UBM modelu a i-vektor extraktoru.

Jednotlivé fáze jsou ošetřeny proti nechtěnému přepsání napočítaných dat nebo modelů při znovuspuštění stejné fáze. Dále je ošetřeno spouštění jednotlivých fází výpočtů v nesprávném pořadí. Pokud například chybí připravená data nebo model, spouštěná fáze se vůbec neprovede a vypíše se upozornění.

Uživatelské proměnné a jiné pevně nastavené parametry, které jsou dále předávány dílčím skriptům, jsou ve skriptech importovány pomocí nástroje `parse_options.sh` (viz adresář `utils`). Jeho funkcí je zpracovat všechny argumenty volaného skriptu označené standardním unixovým prefixem volby a inicializovat novou lokální proměnou. Do nově vytvořené proměnné uvnitř volaného skriptu je pro každou volbu následně přiřazena hodnota předávaného parametru.

Výpočty, které se opakují pro různé typové množiny mluvčích (train, test...), jsou prováděny ve `for` cyklech pro snadnou rozšiřitelnost. Dodatečně zde byla například implementována množina mluvčích pro zápis (enroll).

Implementace `run.sh` je postavena z velké části na již připravených skriptech v adresáři `sid` (viz obr. 11), které jsou součástí nástroje KALDI. Několik podpůrných skriptů (modulů) pro jednotlivé výpočty bylo připraveno do adresáře `local`. Jedná se například o skripty pro přípravu a formátování dat, u kterých první slovo v názvu souboru označuje jméno zpracovávané databáze. Jako pomocné nástroje byly připraveny mj. `trial` skripty pro generování seznamů, které slouží jako podklad pro výpočet skóre. Jako samostatné skripty jsou také implementovány jednotlivé metody pro výpočet skóre, vyhodnocení chybových měr a výpočty verifikační fáze.

Dílčí procedury specifické pro implementaci s databázemi SPEECON a SpeechDat jsou umístěny ve složce `local`, viz následující seznam.

- `speecon_data_prep.sh` nebo `speechdat_data_prep.sh`
- `speecon_create_list.pl` nebo `speechdat_create_list.pl`
- `format_data.sh`
- `trial_list.sh`
- `cosine_scoring.sh`
- `lda_scoring.sh`
- `plda_scoring.sh`
- `compute_identify_err.sh`
- `verification_list.sh`
- `compute_FAR_FRR_EER.sh`

Seznam je seřazen podle pořadí, v němž jsou skripty volány v jednotlivých fázích, jejich implementace bude vysvětlena v podkapitolách STAGE 0 a STAGE 4–5. V seznamu nejsou uvedeny skripty, které jsou již dostupné jako součást KALDI (adresář `sid`).

4.2.1 STAGE 0 – Příprava a formátování dat

Při řešení každé úlohy je nutné nejprve připravit data. Daty se rozumí konkrétní seznam (v textové podobě), který odkazuje na uložené soubory nahrávek, se kterými se bude pracovat. Dále se specifikují například vazby mezi jednotlivými mluvčími a jejich promluvami.

Data se obvykle rozdělují podle typu použití. Běžné je rozdělení na skupiny do samostatných adresářů:

- `train` – data pro trénování modelů,
- `test` – testovací (rozpoznávaná) data.

Většinou je skupin dat více a vždy záleží na typu úlohy. V této implementaci je struktura podadresářů vstupních dat rozdělena podle typu množiny mluvčích, viz obr. 11. Kromě typických adresářů `train` a `test` lze volitelně ještě používat adresář `enroll` pro oddělení množiny zapsaných mluvčích od trénovacích dat. Adresář `data/local` ve vstupní datové struktuře z hlediska trénování modelů slouží k prvotní přípravě dat z indexového souboru právě zpracovávané databáze[3]. V dalších fázích jsou zpracovávána data až z podadresářů `train` a `test`, kde už musí být připravena v požadovaném formátu.

Aby bylo možné v následujících fázích například napočítat příznaky řeči a trénovat statistické modely, je nutné připravit seznamy pro práci s nahrávkami do přesně definovaného formátu. V dokumentaci projektu [9] je popsáno, které soubory a v jakém formátu jsou očekávány v každé ze složek, jež mají být dále nezávisle zpracovávány. Mezi nejdůležitější patří soubor `wav.scp`, ve kterém je specifikováno, jak mají další použité nástroje jednotlivě zpracovat každou nahrávku. Zde je možné specifikovat předzpracování zvukového signálu externím programem, například upravit vzorkovací frekvenci nebo specifikovat použité kvantování vytvořených nahrávek, pokud nejsou tyto informace v hlavičce souboru a používají se nekomprimovaná RAW data, jelikož knihovna KALDI očekává souborový formát `wav`.

Dále jsou důležité ještě soubory `spk2utt` a `utt2spk`. První definuje přiřazení promluvy konkrétnímu mluvčímu a druhý definuje přiřazení mluvčích k jednotlivým promluvám.

Seznámení s oběma databázemi je uvedeno v kapitole 5 a je nutné pro úplné pochopení následujících odstavců v této podkapitole. Proměnné pro přípravu dat v první části `run.sh` slouží k výběru použité skupiny mluvčích (all, OFFICE...) a k výběru množiny nahrávek pro každého mluvčího. Nahrávky některých mluvčích je možné vyřadit z výběru mluvčích pomocí regulárního výrazu v proměnné `EXCEPT_PATTERN`. Množina nahrávek pro každého mluvčího je vybírána pomocí regulárních výrazů (příkazem `grep`) ze seznamu odvozeného z názvů souborů nahrávek. Počet zapsaných mluvčích (enroll) lze redukovat proměnnou `REDUCE_FACTOR` na $1/n$ mluvčích pro uplatnění verifikace podle nastaveného prahu `THRESHOLD`. Pro hodnotu $n = 1$ nebude redukce provedena a výpočet EER nemá význam, ale chybu identifikace je možné vyhodnotit pro plný počet mluvčích.

Hlasy mluvčích u databáze SPEECON jsou nahrány pomocí čtyř typově různých mikrofonů (viz 5.2.6) a jednotlivé kanály jsou označeny CS0 až CS3. Označení kanálu je přímo v názvech souborů nahrávek jako extenze. Nastavením proměnné `CHANNEL` je možné vybrat kanál nahrávky CS0 až CS3, případně alternaci (hodnota alt) těchto čtyř kanálů během výběru nahrávek z nastavené omezující množiny. U databáze SpeechDat je k dispozici pochopitelně pouze jeden kanál označený jako CSA. Naproti tomu je možné na základě jazyka vybrat, která část databáze SpeechDat bude použita.

Převzorkování signálu a změna hloubky kvantizace je možná pouze u databáze SPEECON, a to pomocí uživatelské proměnné.

Pro hlubší pochopení implementace je v následujících čtyřech odstavcích popsána funkcionalita jednotlivých skriptů v adresáři `local`.

[3] Indexový soubor je součástí databáze a umožňuje snadno v databázi nahrávek vyhledávat.

Úkolem skriptu `speecon_data_prep.sh` nebo `speechdat_data_prep.sh` (podle typu databáze) je připravit data z indexového souboru přiloženého k databázi mluvčích. Nejprve je z požadované množiny mluvčích (HOME, OFFICE ...) do souboru `list.scp` vypreparován seznam identifikátorů nahrávek. Pokud nemají být někteří mluvčí zahrnuti ke zpracování, jsou jejich nahrávky ze seznamu vyjmuty regulárním výrazem. Následně je vytvořen nový redukovaný seznam mluvčích výběrem n-tého mluvčího, kde n je redukční faktor. V dalším kroku je provedena selekce požadovaných nahrávek od každého mluvčího z obou seznamů (`spk_list` a `spk_list_reduced`) a tímto jsou vytvořeny finální seznamy nahrávek (`list_train`, `list_enroll` a `list_test`) k dalšímu zpracování. Identifikátory jsou odvozeny z názvů souborů nahrávek (viz podkapitola 5.1); například u databáze SPEECON je každý mluvčí identifikován řetězcem `SANNN` a jeho nahrávka řetězcem `SANNNCCC`, analogické je řešení i u databáze SpeechDat. V posledním kroku jsou pomocí `speecon_create_list.pl` připravena data do požadovaného formátu pro KALDI (formát WAV).

Vstupní parametry skriptu `speecon_create_list.pl` jsou: seznam identifikátorů nahrávek, použitý kanál a parametry pro případné převzorkování signálů. Konfigurace v souborech s extenzí `scp` pro předzpracování databáze je ve tvaru, který ukazuje následující výpis.

```
SANNNCCC sox -t raw -r 16000 -e signed-integer -b 16 -c 1
/path-to-record/SANNNCCC.CS0
-t wav -r 8000 -e signed-integer -c 1 -|
```

Jedná se o příklad, ve kterém je signál převzorkován na frekvenci 8 kHz, výstup je uložen do formátu WAV jedním kanálem a následně je poslán na standardní výstup. Pro úpravu surových dat a případné převzorkování se tímto způsobem používá program `SoX` (Sound eXchange), který každou nahrávku zpracuje samostatně. Dále jsou v rámci skriptu z identifikátorů nahrávek sestaveny seznamy `spk2utt` a `utt2spk`. U obou databází je totiž součástí názvu nahrávky i pořadové číslo mluvčího. Oproti skriptu `speechdat_create_list.pl` je zde navíc implementováno střídaní kanálů CS0 až CS3.

Úlohou skriptu `format_data.sh` je pouze rozřadit připravené seznamy do adresářů podle typu dat (train, test...) a nastavit názvy souborů tak, jak je očekávají nástroje KALDI.

Skriptu `trial_list.sh` jsou předány dva parametry (cesty k seznamům), a to seznam identifikátorů všech mluvčích a redukovaný seznam mluvčích. Redukovaný seznam představuje množinu referenčních mluvčích (enroll) a plný seznam množinu testovaných mluvčích. Ke každému identifikátoru z referenční množiny je přiřazen identifikátor z testovací množiny, výstupem je seznam o dvou sloupcích. Pokud bude v obou vstupních seznamech identifikátorů stejný počet mluvčích (identifikace v uzavřené množině), jedná se vlastně o dvoučlennou variaci s opakováním o velikosti n mluvčích, tedy $V'(2, n) = n^2$. V tomto případě bude výstupem soubor `trials` o velikosti n^2 řádků, který bude použit ve fázi STAGE 4 při výpočtu skóre.

4.2.2 STAGE 1 – Výpočet MFCC a VAD

Výpočet příznaků řeči MFCC a VAD (Voice Activity Detection) musí předcházet fázi STAGE 2 a 3, jelikož veškeré dílčí procedury použité v těchto fázích s napočítanými MFCC a VAD pracují.

Během této fáze jsou vypočteny a uloženy pouze základní MFCC koeficienty, derivace koeficientů (delta a delta-delta) jsou vždy počítány až těsně před použitím v každé dílčí proceduře, jež s nimi pracuje.

Pro výpočet MFCC koeficientů byly převzaty výchozí parametry z KALDI, které jsou optimalizovány pro vzorkovací frekvenci 16 kHz, viz následující seznam:

- délka okna 25 ms,
- posun okna o 10 ms,
- počet mel-frekvenčních trojúhelníkových filtrů 23.

Detekce hlasové aktivity je v KALDI (viz `compute-vad`) řešena jednoduchým způsobem na bázi logaritmu energie řečového segmentu vypočtené pro každý segment zvlášť již během výpočtu MFCC. Podle stanoveného prahu je rozhodováno, zda se jedná o segment řeči, a následně je pro každý segment uložena maskovací informace.

Z důvodů větších nároků na diskový prostor jsou MFCC koeficienty a VAD maskovací informace, jež jsou v binární podobě, ukládány mimo adresářovou strukturu `v1`. Odkaz na ně je poté pro další fáze k dispozici ve stejné složce, ve které jsou připraveny seznamy nahrávek, viz 4.2.1. Z tohoto důvodu jsou si fáze STAGE 0 a 1 blízké, i tuto fázi je tudíž možné chápat jako přípravu dat.

Dílčí procedury `make_mfcc.sh` a `compute_vad_decision.sh` nebylo nutné implementovat, jelikož jsou již součástí sady nástrojů KALDI. V rámci těchto skriptů je vyřešeno i paralelní zpracování nahrávek. Každá samostatně pracující úloha zpracovává pouze část nahrávek a její výstup je ukládán do vlastního samostatného souboru.

4.2.3 STAGE 2 – Trénování UBM a i-vektor extraktoru

Implementace i-vektorového systému (viz 3.1.4) je v KALDI řešena ve třech následujících krocích.

1. Příprava UBM s diagonální kovarianční maticí.
2. Přetrénování na UBM s plnou kovarianční maticí.
3. Natrénování extraktoru i-vektorů (prostor celkové variability).

Dle výsledků v [8] je s plnou kovarianční maticí při stejném počtu GMM komponent dosahováno výrazně lepších výsledků. Lze říci, že pokud bude existovat závislost mezi jednotlivými příznaky (uvnitř příznakového vektoru), bude pro popis modelu třeba méně GMM komponent. V každém ze tří kroků je používán EM algoritmus [9], a tudíž se jedná o výpočetně nejnáročnější fázi.

Z parametrů uživatelsky nastavitelných v `run.sh` jsou v této fázi nejdůležitější následující:

- počet GMM komponent pro přípravu UBM modelu,
- dimenze i-vektorů (extrahovaných ve fázi STAGE 3).

Ve složce `exp` (viz obr. 11) jsou po provedení výpočtů k dispozici podadresáře natrénovaných modelů, u kterých je z hlediska fáze STAGE 3 (například dodatečné extrakce i-vektorů nových dat) důležitý především model extraktoru (složka `extractor`) a dále UBM s plnou kovarianční maticí (složka `full_ubm`). Modely jsou ukládány v binární podobě a pro případné zpracování programem MATLAB lze uložit UBM model i v textové podobě.

Dílčí procedury (viz následující seznam) jsou součástí sady nástrojů KALDI a nebudou zde podrobně popisovány.

- `train_diag_ubm.sh`
- `train_full_ubm.sh`
- `train_ivector_extractor.sh`

Z hlediska paralelního zpracování je zde také standardně používáno rozdělení na úlohy (jobs), které jsou předávány jednotlivě ke zpracování do fronty SGE. U 1. a 3. kroku lze rozdělit výpočty dále ještě mezi vlákna (threads), avšak počet nastavených vláken nesmí překročit počet jader procesoru, jež je součástí systému SGE jako jedna výpočetní jednotka.

4.2.4 STAGE 3 – Extrakce i-vektorů

Za předpokladu, že je již připraven UBM a extraktor (viz předchozí fáze), jsou z napočítaných MFCC pomocí UBM připraveny postačující statistiky (posteriors), následně jsou z nich extrahovány i-vektory pro jednotlivé nahrávky u kterékoli množiny dat (train, enroll, test). Tento postup zachycuje obr. 8 v kapitole 3.

Dílčí procedura `extract_ivectors.sh` je také součástí sady nástrojů KALDI. V její poslední části je řešeno průměrování i-vektorů všech nahrávek daného mluvčího (u všech mluvčích), aby bylo možné mezi sebou porovnávat přímo i-vektory mluvčích.

4.2.5 STAGE 4 – Výpočet skóre pro zvolené metody

Napočítané skóre je ukládáno do adresáře `scores` (viz obr. 11), a to do samostatných souborů podle použité metody výpočtu skóre. Například pro metodu PLDA je skóre uloženo do textového souboru `plda_scores`. Dále jsou zde, podobně jako v ostatních podadresářích složky `exp`, chybové a protokolové výstupy jednotlivých výpočtů. Skóre je formátováno do 3 sloupců, kde první dva sloupce obsahují identifikátory mluvčích (enroll/train a test) a třetí sloupec je hodnota skóre vzájemné podobnosti i-vektorů těchto dvou mluvčích.

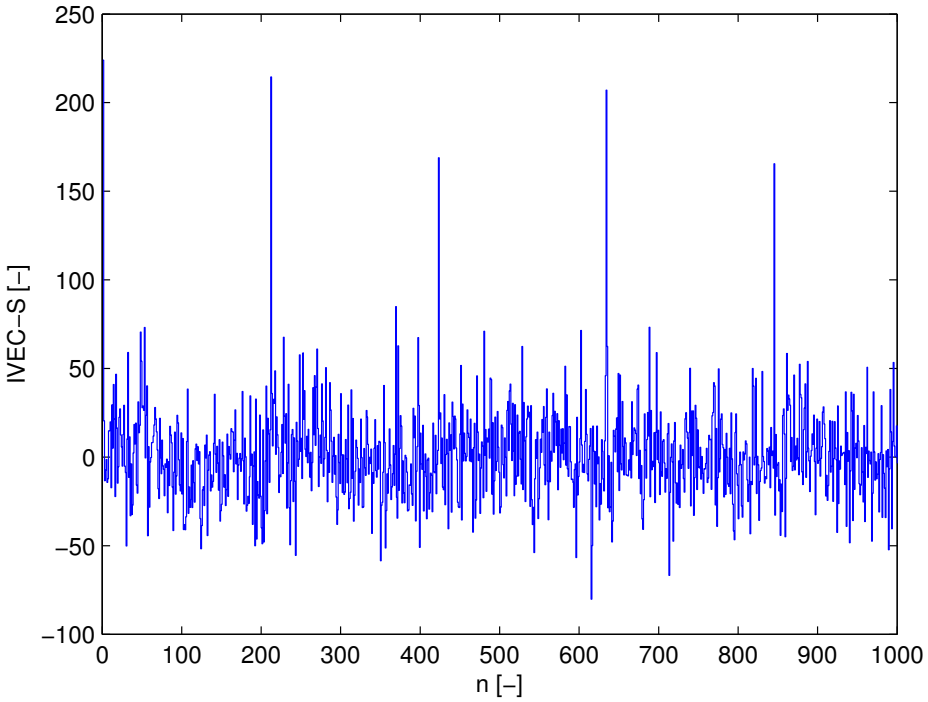
Metody pro výpočet skóre lze jednotlivě zapnout nebo vypnout v uživatelské části `run.sh`. Výběr pouze některých metod je vhodný například v situaci, kdy při některých výpočtech není možné použít LDA transformaci. Typická je situace při nízké dimenzi i-vektorů před transformací.

Použití jednotlivých metod je na sobě nezávislé (viz následující seznam). Implementace každé metody ve složce `local` je vysvětlena následně.

- Výpočet kosinové vzdálenosti.
- Transformace LDA s následným výpočtem kosinové vzdálenosti.
- Výpočet skóre pomocí PLDA modelu.

Ve skriptu `cosine_scoring.sh` je na základě souboru `trials` ohodnocena vzájemná podobnost každé dvojice mluvčích, respektive jejich i-vektorů. Z obou vektorů jsou pomocí knihovní funkce `ivector-compute-dot-products` spočteny skalární součiny (coby skóre) pro všechny porovnávané dvojice. Malý výřez hodnot skóre z výstupního souboru je zanesen do grafu 12. Hodnota skóre tedy není normalizována součinem délek vektorů (viz vztah (17)), nicméně v KALDI projektu `sre08` je výpočet označen jako kosinová vzdálenost. U všech tří skriptů pro výpočet skóre (i `lda_scoring.sh` a `plda_scoring.sh`) je postup odvozen z projektu `sre08`.

U `lda_scoring.sh` jsou pro výpočet transformační matice použity i-vektory nahrávek z množiny trénovacích dat a dále je použit seznam přiřazení mluvčích k jednotlivým promluvám `spk2utt`. Poté mohou být transformovány i-vektory referenčních (enroll) a testovaných mluvčích. Následně jsou spočteny opět skalární součiny transformovaných i-vektorů. Výstupem je podobně jako u `cosine_scoring.sh` soubor, který obsahuje tři sloupce. První dva jsou stejné jako u souboru `trials` a třetí sloupec je napočítané skóre. Na chybovost systému mají vliv také vstupní parametry, jako je dimenze po transformaci a kovarianční faktor. Kovarianční faktor dle [9] (viz `ivector-compute-lda`) určí,



Obr. 12 Skóre jako skalární součin i-vektorů (IVEC-S) pro n vyhodnocení. Příklad identifikace v uzavřené množině pro 210 mluvčích. Prvních 5 mluvčích (výřez) bylo porovnáno nejprve se sebou samým a poté s ostatními.

zda bude normalizace provedena uvnitř tříd kovariance pro hodnotu parametru 1, nebo u celkové kovariance pro hodnotu 1. Pro hodnoty mezi 0 a 1 se interpoluje.

Předávání vstupních parametrů u `plda_scoring.sh` je podobné jako u implementace LDA. Místo výpočtu transformační matice je natrénován PLDA model. Skalární součin není počítán přímo jako v předchozích dvou případech, ale pomocí nástroje `ivector-plda-scoring` z knihovny [9]. Rozdíl v hodnotách skóre proti předchozím dvěma případům (viz graf 12) je v tom, že střední hodnota přes všechny hodnoty skóre není nulová, jelikož je skóre počítáno jako logaritmus pravděpodobnosti.

Cílem skriptu `compute_identify_err.sh` je nalézt chybu identifikace za předpokladu, že je k dispozici plný i redukovaný seznam mluvčích a napočítané skóre. Pro každého testovaného mluvčího (spk_true) je nalezeno nejvyšší skóre v porovnání se všemi referenčními mluvčími a následně je vyhodnoceno, zda testovaný mluvčí byl totožný s referenčním. Pokud nenastala shoda (nontarget), je inkrementován čítač počtu chyb. Pokud se hledá chyba identifikace v otevřené množině, musí být testovaný mluvčí také na seznamu referenčních, v opačném případě se nejedná o chybu identifikace a čítač není inkrementován. Nakonec je chyba P_{err} vypočtena dle vztahu (3).

4.2.6 STAGE 5 – Verifikace

Cílem verifikační fáze je otestovat přesnost klasifikace mluvčího na základě chybové míry EER. Dále tato fáze umožňuje nasimulovat chování verifikačního systému pro konkrétní práh za předpokladu, že byl mluvčí správně identifikován. Implementace byla rozdělena do dvou skriptů ve složce `local`, viz dále.

Ve skriptu `verification_list.sh` je identifikace mluvčího implementována stejným způsobem jako u `compute_identify_err.sh` (viz sekce 4.2.5), ale na vstupu je předána ještě hodnota prahu. Práh je porovnán s hodnotou skóre právě identifikovaného mluvčího (spk_true), který měl skóre nejvyšší (první kritérium). Druhým kritériem je porovnání, zda byl identifikovaný mluvčí shodný s referenčním (target) či nikoliv (nontarget). Podle těchto dvou kritérií je rozhodnuto, který ze čtyř případů nastal, viz krátký výpis verifikačního seznamu.

```
126.1272 nontarget - reject OK
126.7496 nontarget - reject OK
128.1317 target - reject ERR
128.1365 nontarget - reject OK
128.6234 target - reject ERR
129.9324 nontarget - reject OK
--------------------------------------
130.2905 nontarget - accept ERR
133.3013 nontarget - accept ERR
133.4881 target - accept and identify OK
133.7035 target - accept and identify OK
134.0013 target - accept and identify OK
```

Výpis má ukázat příklad malé části seřazených hodnot skóre pro nastavený práh 130. U některých mluvčích došlo k chybnému zamítnutí (reject ERR) a naopak u některých došlo k neoprávněnému přijetí (accept ERR). V případě, že byla identifikace mluvčího chybná, je ve fázi verifikace chybně identifikovaný mluvčí v podstatě vyřazen z okruhu referenčních mluvčích. Chyba identifikace mohla vzniknout u skóre nad prahem, nebo také pod prahem, ale obvykle je chyba identifikace výrazně nižší než chyba verifikace.

Skriptu `compute_FAR_FRR_EER.sh` je předán ohodnocený verifikační seznam. Na základě vztahů (4) a (5) jsou vypočteny chybové míry FAR a FRR. Pro výpočet EER postačí hodnoty skóre a k nim přiřazená informace, zda byl mluvčí pravý nebo nikoliv (target/nontarget), následně je pro výpočet chyby použit nástroj `compute-eer`.

5 Experimentální část

Experimentální část se zabývá identifikací mluvčího v uzavřené nebo otevřené množině mluvčích, identifikace je založena na GMM s i-vektorovou reprezentací mluvčích (viz implementace úlohy `v1`). V případě otevřené množiny je počítána také chyba verifikace. Oba chybové ukazatele jsou počítány a vyhodnoceny v závislosti na různých vstupních parametrech úlohy a jednotlivé série experimentů byly testovány na databázích SPEECON [16] a SpeechDat [17]. Některé experimenty byly provedeny pouze pro jednu z databází, pokud pro druhou z nich experiment pozbýval na významu. V této části jsem mohl nasimulovat chování modelu úlohy `v1` na malém vzorku populace, nicméně pro otestování úlohy při použití v menších reálných aplikacích (viz kapitola 1) jsou data plně vyhovující.

5.1 Použité databáze

Pro experimenty nad širokopásmovými daty byla zvolena česká databáze dospělých mluvčích SPEECON [16], databáze čítá 590 mluvčích. Pro každého mluvčího je k dispozici přibližně 15 minut nekomprimovaných širokopásmových nahrávek a je tedy možné rozličně kombinovat množinu trénovacích a testovacích promluv z více různých kategorií nahrávek. Možné kategorie jsou například foneticky bohaté věty nebo samostatné číslovky. Širokopásmové nahrávky jsou vzorkovány frekvencí 16 kHz a kvantovány na 16 bitů. Seznam všech nahrávek databáze SPEECON lze rozdělit do několika skupin podle typu místa, kde bylo nahrávání uskutečněno. K dispozici jsou tyto skupiny:

- ENTERTAINMENT,
- OFFICE,
- PUBLIC PLACE,
- CAR.

Skupina ENTERTAINMENT (domácí prostředí) obsahuje 33 mluvčích. Z této skupiny byly použity pouze nahrávky mluvčích, u kterých nebylo zapnuté rádio na pozadí, což by představovalo další zkreslující element. Skupina mluvčích OFFICE, tedy nahrávky z kancelářských prostor, obsahuje nejméně rušivých vlivů; patří do ní 210 mluvčích. Zbylé dvě skupiny mluvčích obsahují mnoho šumu a cizích zvuků na pozadí. U nahrávání v automobilu (CAR) je zde výrazný korelovaný šum, který představuje zapnutý motor. Nahrávky na veřejném prostranství (PUBLIC PLACE) obsahují zase mnoho cizích náhodných zvuků. Pro experimenty je vhodná zejména skupina OFFICE, jelikož počet mluvčích odpovídá necelé polovině z celkových 590.

Název souboru každé nahrávky databáze SPEECON je ve formátu `SANNNCCC.CSO`, kde `NNN` je pořadové číslo mluvčího (000 až 589) a `CCC` je kód korpusu. Kód označuje obsah (typ) promluvy (číslovky, věty, samostatná slova...) a extenze souboru značí kanál nahrávky.

Dle dokumentace databáze SPEECON a poslechu byly vyřazeny některé příliš zarušené nahrávky mluvčích. Jednalo se o mluvčí s indexem SA568–589. Po vyřazení těchto nahrávek je v experimentech skupina dále označována jako *all* a obsahuje 568 mluvčích.

Pro experimenty s daty v telefonní kvalitě byla vybrána česká databáze SpeechDat [17], jež obsahuje promluvy od 1 052 mluvčích. Nahrávky jsou nasbírány převážně z pevných telefonních sítí. Pro každého mluvčího je k dispozici v průměru přibližně 2,7 MB nahrávek. Při vzorkovací frekvenci 8 kHz a 8bitovém kvantování to odpovídá více než 5 minutám záznamu pro každého. Rozdělení mluvčích podle pohlaví je rovnoměrné, stejně jako u databáze SPEECON. Zastoupení každého pohlaví je přesně 50 %.

Nejčastější místa volajících nebo volaných účastníků, z nichž jsou nahrávky pořízeny, jsou:

- HOME,
- OFFICE,
- BOOTH.

Skupina účastníku HOME obsahuje 819 mluvčích, skupina OFFICE 137 a skupina mluvčích nahrávaná v telefonních budkách (BOOTH) 21. Skupina všech 1 052 mluvčích je označena *all*.

Název souboru každé nahrávky u databáze SpeechDat je podobně jako u databáze SPEECON ve formátu `A3NNNNCC.CSA`, kde `NNNN` je pořadové číslo mluvčího (0000 až 1051) a `CC` je kód korpusu (typ promluvy).

Z hlediska verifikace mluvčího byl u obou databází stanoven model simulace tak, aby byla verifikační fáze podobná situaci, ve které by mohl být systém realizován (například na lince technické podpory, viz kapitola 1). Simulací nad databází SpeechDat byla modelována situace, ve které na technickou podporu pravidelně telefonuje 1 052 (celkový počet mluvčích databáze – UBM) zákazníků, část z nich je zapsána jako referenční množina mluvčích (enroll) a je během hovoru po dostatečně dlouhé době autentizována. Je tedy s určitou chybou rozhodnuto, zda je právě volající uložen v databázi mluvčích nebo nikoli. Postup verifikace je vysvětlen v rámci implementace v podkapitole 4.2.6.

5.2 Výchozí nastavení experimentů

Pokud není uvedeno jinak, byla u experimentů použita pouze data trénovací a testovací. Od jednotlivých mluvčích jsou použity buď všechny nahrávky, nebo některá ze dvou redukovaných množin nahrávek. Menší ze dvou množin obsahuje pouze číslovky (zde označena jako NUMERIC), kdežto větší z nich obsahuje ještě foneticky bohaté věty (označena jako NUMERIC+SENTENCE).

U množiny nahrávek NUMERIC z databáze SPEECON [16] byly ve výběru trénovacích dat zvoleny 3 promluvy typu CC1–3[1] a ve výběru testovacích dat 1 promluva typu CC4. U databáze SpeechDat [17] byly zvoleny 2 promluvy (C1–2) pro trénování a 1 (C4) pro testování. Na trénování GMM modelu tedy u obou databází připadá přibližně 15 namluvených číslovek, pro testování jich je 5.

Výběr trénovacích i testovacích dat u množiny NUMERIC+SENTENCE obsahuje přibližně z jedné poloviny číslovky a z druhé foneticky bohaté věty. Do trénovacího výběru nahrávek z databáze SPEECON byly použity 3 promluvy číslovek CC1–3 a 3 promluvy obsahující krátké věty S01–3. Do testovacího výběru byly zařazeny promluvy CC4 a S04. Pro databázi SpeechDat byly analogicky vybrány promluvy C1–2 a S1–3 pro trénování a jako testovací promluvy C4 a S4. Právě rozšířením směsi nahrávek od každého mluvčího ze samotných číslovek o věty bylo dosaženo výrazně lepších výsledků. Pro každého mluvčího je délka směsi (číslovky a věty) promluv u obou databází v průměru 30 sekund u trénovacích dat a 10 sekund u testovacích dat.

[1] Typ promluvy je odvozen z názvu souboru, viz 5.1.

V experimentech, v nichž byly použity všechny nahrávky, byly do výběru testovacích promluv zařazeny věty S01–30 pro SPEECON a věty S0–9 pro SpeechDat. Zbylé promluvy byly použity pro trénování modelu. Zařazení promluv z obou databází do skupin přehledně ukazuje tab. 1.

U většiny experimentů byly u databáze SPEECON použity nahrávky blízkým mikrofonem (CS0 – výchozí nastavení). Jedná se o nahrávky nejvyšší dostupné kvality, které byly získány v reálném prostředí. S těmito nahrávkami bylo možné dosáhnout nejvyšší (pravděpodobně hraniční) přesnosti rozpoznávaní mluvčího.

	SPEECON	SpeechDat
NUMERIC	CC1–3, CC4	CC1–2, C4
NUMERIC+SEN.	CC1–3 + S01–3, CC4 + S04	C1–2 + S1–3, C4 + S4
Všechny promluvy	Zbytek, S01–30	Zbytek, S0–9

Tab. 1 Použité množiny trénovacích a testovacích nahrávek v experimentální části

U většiny experimentů, pokud není uvedeno jinak, je UBM model a extraktor i-vektorů natrénován z množiny všech mluvčích. Zapsaných (referenčních) mluvčích je obvykle polovina, aby bylo možné vyhodnotit chybu verifikace EER (viz výše). Chyba identifikace mluvčího v otevřené množině je vyhodnocena též z poloviního počtu uvedených mluvčích. Pro testovací množinu mluvčích je použit plný počet mluvčích, jako u trénování modelu. V reálných podmínkách by rozdíl počtu referenčních a testovaných mluvčích byl pravděpodobně větší, ale pro výpočet chyby identifikace není vhodné nastavit nízký počet mluvčích v důsledku malého rozlišení chyby. Počty mluvčích u jednotlivých sérií experimentů jsou v každé tabulce označeny ve tvaru zlomku, a to jako počet zapsaných (referenčních) ku počtu testovaných (rozpoznávaných) mluvčích.

Chyba identifikace P_{err} (3) nebo EER (6) v případě verifikace je v tabulkách vyhodnocena na základě skóre z porovnání dvou i-vektorů, kde i-vektor reprezentuje na jedné straně mluvčího z trénovací množiny promluv a na druhé straně mluvčího z testovací množiny. Porovnání i-vektorů z matematického hlediska je popsáno v podkapitole 3.1.4 a popis implementace pro výpočet skóre je uveden v sekci 4.2.5. Tento typ výpočtu chyby je v tabulkách označen zkratkou IVEC. Pokud je použita lineární diskriminační analýza, je použita zkratka LDA. V případě vyhodnocení skóre pomocí modelu PLDA je použita zkratka PLDA (viz 3.2.2).

Pokud to bylo možné, byly vypočteny i chyby u metody LDA, nicméně z důvodu lepší přehlednosti a zachování jednotného rozvržení tabulek do šířky nebyly hodnoty LDA do tabulek většinou vyneseny. Chyba po vyhodnocení metodou LDA totiž po optimálním nastavení parametrů `LDA_COVAR_FACTOR` a `LDA_DIM` ve většině případů odpovídala hodnotě mezi IVEC a PLDA. V reálných aplikacích je většinou vhodnější použít metodu PLDA.

Platí pravidlo, že čím více mluvčích bude zařazeno do experimentu, tím větší bude rozlišovací schopnost pro zjištění chybovosti systému. U experimentů, kde byla požadována co největší rozlišovací schopnost, byly použity vybrané promluvy všech mluvčích. Některé experimenty, u kterých se chyba identifikace nebo verifikace blíží nule, jsou vlivem nízkého rozlišení vyhodnoceny v tabulkách s „nulovou“ chybou. Bezchybná identifikace se obecně vyskytuje častěji při malém počtu mluvčích.

Jednotlivé experimenty mají pro různé testované vstupní proměnné v tabulkách zástupce s výchozími parametry modelu a parametry pro výpočet příznaků. Výchozí parametry pro experimenty byly stanoveny buď na základě předchozích experimentů autorů KALDI a nebo na základě experimentálního ověření v této implementaci. V druhém

případě byla zvolena taková hodnota, při které docházelo k co nejnižší chybě identifikace. Nejdůležitější vstupní parametry a jejich výchozí hodnoty jsou shrnuty v tab. 2. Parametry, které se týkají přípravy dat a výpočtu skóre (i chybových měr) jsou podrobněji vysvětleny v kapitole 4.

Název parametru	Hodnota parametru
Dimenze MFCC vektoru	20
Počet GMM komponent	64
Dimenze i-vektorů	400
LDA zmenšení dimenze	250
LDA kovarianční faktor	0,05
Kanál nahrávek (SPEECON)	CS0

Tab. 2 Výchozí nastavení parametrů (vstupní proměnné úlohy)

Následuje výčet parametrů nebo veličin, u kterých je sledována chybovost pro různé hodnoty. Jednotlivé série experimentů jsou popsány v následujících podkapitolách.

- Počet GMM komponent (5.2.1).
- Velikost dimenze i-vektorů (5.2.2).
- Vzorkovací frekvence nahrávek (5.2.3).
- Velikost dimenze MFCC vektoru (5.2.4).
- Skupina mluvčích (počet mluvčích) (5.2.5).
- Použitý kanál nahrávek (CS0–CS3) u databáze SPEECON (5.2.6).
- Parametry při generování transformační matice pomocí LDA (5.2.6).
- Různé množství testovacích promluv od jednoho mluvčího (5.2.7).
- Hodnota prahu pro vyhodnocení chyby FAR a FRR (5.2.8).

5.2.1 Optimalizace počtu GMM komponent

Pokud je model natrénován obecně s nízkým počtem GMM komponent, nebo je naopak přetrénován s vysokým počtem, zvyšuje se chyba identifikace i EER. Pro zjištění optimálního nastavení počtu GMM komponent pro další experimenty byly otestovány obě databáze v rozsahu 32–512 komponent. Při trénování modelu se u testů nad 512 komponent už vyskytovaly na dostupném hardwaru problémy s nedostatkem paměti RAM.

Jako optimální se pro obě databáze s plným počtem mluvčích na množině nahrávek NUMERIC+SENTENCE dle výsledků (viz tab. 3 a 4) jeví 64 GMM komponent. Optimum pro toto typické množství a kvalitu dat v následujících experimentech tedy leží někde mezi 64 a 128 GMM komponentami.

Nejvyšší přesnosti bylo dosaženo při použití všech nahrávek přes celou databázi mluvčích, kde do množiny testovacích promluv byly zařazeny všechny foneticky bohaté věty (viz tab. 1). Chyba identifikace byla ve všech případech 0 % a chyby EER pro obě databáze jsou vyneseny v tabulkách 5 a 6. Délka testovaných promluv z celkové délky všech nahrávek je přibližně 15 % pro SPEECON a 25 % pro SpeechDat.

Nutno podotknout, že u aplikace v reálném provozu takovéto množství dat nebude k dispozici, a to především u právě zpracovávaných dat. Navíc při porovnání hodnot mezi tabulkami z množinou nahrávek NUMERIC+SENTENCE a množinou všech nahrávek od každého mluvčího je zřejmé, že výrazným zvýšením množství dat už nedochází ke stejně výraznému zvýšení přesnosti rozpoznávání.

GMM komponent	IVEC P_{err} [%]	PLDA P_{err} [%]
32	0,35	0
64	0	0
128	0	0
256	0	0
512	0	0,35
GMM komponent	IVEC EER [%]	PLDA EER [%]
32	2,82	1,05
64	**1,40**	**0**
128	2,11	0,70
256	2,46	1,05
512	3,52	3,88

Tab. 3 Chybovost (SPEECON – NUMERIC+SENTENCE) pro různý počet GMM komponent, 284/568 mluvčích

GMM komponent	IVEC P_{err} [%]	PLDA P_{err} [%]
32	0,19	0
64	0,19	0
128	0,38	0,19
256	0,19	0
512	0,19	0
GMM komponent	IVEC EER [%]	PLDA EER [%]
32	4,76	1,90
64	**4,00**	**1,33**
128	4,38	1,52
256	4,19	2,28
512	4,76	3,04

Tab. 4 Chybovost (SpeechDat – NUMERIC+SENTENCE) pro různý počet GMM komponent, 526/1052 mluvčích

GMM komponent	IVEC EER [%]	PLDA EER [%]
64	0,70	0,35
128	0,35	0,35
256	0,35	0,35

Tab. 5 Chybovost (SPEECON – všechny promluvy) pro různý počet GMM komponent, 284/568 mluvčích

Pro představu uvedu výpočetní náročnost na SGE (Sun Grid Engine) při použití všech dostupných nahrávek. Výpočet byl distribuován na 24 jader procesorů. Jednalo se o 4jádrové procesory typu Intel Core i7 s taktovací frekvencí 3,6 GHz. Během měření byl veškerý čas všech procesorů k dispozici pouze pro tuto úlohu. Reálné časy výpočtů jsou uvedeny v tabulce 7.

GMM komponent	IVEC EER [%]	PLDA EER [%]
64	2,85	1,14
128	**2,66**	1,52
256	3,23	**0,76**

Tab. 6 Chybovost (SpeechDat – všechny promluvy) pro různý počet GMM komponent, 526/1052 mluvčích

Databáze	MFCC + VAD	Trénování modelu	Extrakce i-vektorů
SPEECON	84	36	24
SpeechDat	28	21	8

Tab. 7 Reálný čas procesoru v minutách u fáze STAGE 1 až STAGE 3. Jedná se o výchozí nastavení parametrů s výjimkou počtu GMM komponent (256) a při použití všech dostupných nahrávek.

5.2.2 Volba dimenze i-vektorů

Pro snižování velikosti dimenze extrahovaných i-vektorů (viz tab. 8 a 9) byla jako výchozí hodnota použita dimenze 400, jelikož je nastavena jako výchozí proměnná v KALDI (sre08) ve skriptu `train_ivector_extractor.sh`. U telefonních dat se výrazněji zvýšení chybovosti projeví až při velikosti dimenze 100. Typické použití dimenze i-vektoru u databáze SPEECON by dle výsledků bylo 400. Naopak dimenze 100 už je příliš nízká a je tedy mimo rámec i pro použití v databázi SpeechDat.

Dimenze i-vektoru	IVEC P_{err} [%]	PLDA P_{err} [%]
400	0	0
300	0	0
200	0,35	0
100	1,05	0
Dimenze i-vektoru	IVEC EER [%]	PLDA EER [%]
400	1,40	0
300	1,40	0,70
200	2,47	1,76
100	3,91	3,16

Tab. 8 Chybovost (SPEECON – NUMERIC+SENTENCE) pro různé dimenze i-vektoru, 284/568 mluvčích

Dimenze i-vektoru	IVEC P_{err} [%]	PLDA P_{err} [%]
400	0,19	0
300	0	0,19
200	0	0
100	2,28	0,95
Dimenze i-vektoru	IVEC EER [%]	PLDA EER [%]
400	4,00	1,33
300	4,75	1,71
200	4,37	2,66
100	6,03	3,64

Tab. 9 Chybovost (SpeechDat – NUMERIC+SENTENCE) pro různé dimenze i-vektoru, 526/1052 mluvčích

5.2.3 Vliv šířky pásma

Vzorkovací frekvence u databáze SPEECON (viz tab. 10) byla snížena na úroveň telefonní kvality, účelem bylo porovnat chybovost za předpokladu použití stejných dat a stejné výchozí konfigurace modelu. U telefonních nahrávek v mezinárodním standardu a-law se používá logaritmické 8bitové kvantování, které odpovídá přibližně 13bitovému lineárnímu (PCM) kvantování. U experimentu bylo zachováno původní 16bitové kvantování, jelikož z hlediska chyby rozpoznávání mluvčího je předpokládán minimální rozdíl.

Vz. frekvence	IVEC P_{err} [%]	PLDA P_{err} [%]
16 kHz	0	0
8 kHz	0,70	0,35
Vz. frekvence	IVEC EER [%]	PLDA EER [%]
16 kHz	1,40	0
8 kHz	3,54	2,47

Tab. 10 Chybovost (SPEECON – NUMERIC+SENTENCE) po snížení vzorkovací frekvence na úroveň telefonní kvality, 284/568 mluvčích

Jelikož byla zvolená délka trénovacích a testovacích nahrávek u obou databází přibližně stejná, je možné provést pro stejný počet mluvčích porovnání mezi databází SPEECON (16 kHz) a SpeechDat (8 kHz), viz tab. 11. Počet mluvčích databáze SpeechDat byl omezen na prvních 568 z celkového počtu 1 052 tak, aby z obou databází byl použit stejný počet mluvčích. Všechny skupiny mluvčích (HOME, OFFICE...) jsou zastoupeny téměř rovnoměrně. Chyba identifikace byla ve všech čtyřech případech nulová. Je možné si všimnout menší účinnosti metody PLDA u databáze SpeechDat při menším počtu mluvčích, jelikož EER je pro plný počet mluvčích 1,331 %.

Databáze	IVEC EER [%]	PLDA EER [%]
SPEECON	1,40	0
SpeechDat	3,52	2,11

Tab. 11 Porovnání SPEECON a SpeechDat (NUMERIC+SENTENCE), 284/568 mluvčích

5.2.4 Volba dimenze MFCC vektoru

Další experiment ukazuje vliv na chybovost, pokud bude zvolena nižší dimenze MFCC vektoru, než dimenze standardně použitá v KALDI (sre08). Dle [1] v úloze rozpoznávání mluvčího více jak 20 MFCC koeficientů nepřináší výrazné zlepšení přesnosti. Pro obě databáze a samostatně pro chybu identifikace i EER jsou vyneseny chyby do čtyř tabulek (12 až 15). V každé tabulce je porovnáno, jaký vliv má množství dat (trénovacích a testovacích promluv) od každého mluvčího.

U množiny NUMERIC+SENTENCE se chyba identifikace výrazně projevila až při snížení dimenze MFCC na 5. Chyba identifikace je obvykle nižší než chyba verifikace, ale u množiny NUMERIC a použití MFCC dimenze 5 je situace opačná. Použití nízké dimenze MFCC vektoru (5 nebo 10) je nicméně v reálných aplikacích nepřijatelné.

NUMERIC		
MFCC koeficientů	IVEC P_{err} [%]	PLDA P_{err} [%]
20	1,76	1,05
13	3,16	2,46
10	6,33	7,04
5	19,71	25,35
NUMERIC+SENTENCE		
MFCC koeficientů	IVEC P_{err} [%]	PLDA P_{err} [%]
20	0	0
13	0	0
10	0,35	0,35
5	3,16	1,40

Tab. 12 Chyba identifikace (SPEECON) pro různý počet MFCC koeficientů, 284/568 mluvčích

NUMERIC		
MFCC koeficientů	IVEC EER [%]	PLDA EER [%]
20	6,45	3,91
13	8,00	6,85
10	9,02	8,33
5	18,86	18,40
NUMERIC+SENTENCE		
MFCC koeficientů	IVEC EER [%]	PLDA EER [%]
20	1,40	0
13	2,81	0,70
10	4,59	1,41
5	12,73	7,14

Tab. 13 Chyba EER (SPEECON) pro různý počet MFCC koeficientů, 284/568 mluvčích

NUMERIC		
MFCC koeficientů	IVEC P_{err} [%]	PLDA P_{err} [%]
20	4,75	4,18
13	6,08	5,51
10	8,36	7,79
5	26,04	28,70
NUMERIC+SENTENCE		
MFCC koeficientů	IVEC P_{err} [%]	PLDA P_{err} [%]
20	0,19	0
13	0	0
10	0,38	0
5	4,37	1,52

Tab. 14 Chyba identifikace (SpeechDat) pro různý počet MFCC koeficientů, 526/1052 mluvčích

NUMERIC		
MFCC koeficientů	IVEC EER [%]	PLDA EER [%]
20	8,18	7,73
13	9,71	8,04
10	11,62	9,69
5	18,51	18,93
NUMERIC+SENTENCE		
MFCC koeficientů	IVEC EER [%]	PLDA EER [%]
20	4,00	1,33
13	4,94	1,90
10	6,87	1,71
5	14,51	5,21

Tab. 15 Chyba EER (SpeechDat) pro různý počet MFCC koeficientů, 526/1052 mluvčích

5.2.5 Vliv počtu mluvčích

Chyba identifikace při stejné kvalitě dat obecně roste s počtem referenčních mluvčích. Pro otestování závislosti P_{err} na počtu mluvčích byl `REDUCE_FACTOR` roven 1. Jednalo se tedy o identifikaci mluvčího v uzavřené množině. V tabulce 16 (SPEECON NUMERIC) je vidět porovnání dvou stejně velkých skupin s rozdílnou kvalitou nahrávek. Skupina CAR dosahuje nulové chyby IVEC P_{err}, jelikož byly právě z této skupiny vyřazeny silně zarušené nahrávky. Pro množinu promluv NUMERIC+SENTENCES není tab. uvedena, jelikož byla chyba IVEC P_{err} ve všech 5 případech 0 %.

Analogicky je vyhodnocen experiment i pro databázi SpeechDat (tab. 17 a 18), ale zde byly vyhodnoceny pouze 2 nejpočetnější skupiny mluvčích (OFFICE, HOME). Dle výsledků v tab. 18 lze konstatovat, že prostředí HOME je pravděpodobně více zarušené než prostředí jiná.

Skupina	Počet mluvčích	IVEC P_{err} [%]
ENTERTAINMENT	33	0
CAR	73	0
OFFICE	210	2,85
PUBLIC_PLACE	210	4,28
all	568	4,04

Tab. 16 Chyba identifikace (SPEECON – NUMERIC) v závislosti na počtu mluvčích (skupinách)

Skupina	Počet mluvčích	IVEC P_{err} [%]
OFFICE	137	2,91
HOME	819	5,73
all	1052	7,31

Tab. 17 Chyba identifikace (SpeechDat – NUMERIC) v závislosti na počtu mluvčích (skupinách)

Skupina	Počet mluvčích	IVEC P_{err} [%]
OFFICE	137	0
HOME	819	0,24
all	1052	0,19

Tab. 18 Chyba identifikace (SpeechDat – NUMERIC+SENTENCES) v závislosti na počtu mluvčích (skupinách)

5.2.6 Změna kanálu nahrávek

V této sekci se veškeré experimenty budou týkat pouze databáze SPEECON. Jednotlivé kanály CS0 až CS3 byly zaznamenávány současně pro jednotlivé promluvy, viz dokumentace [16]. Následuje výčet jednotlivých typů mikrofonů, které byly použity při nahrávání.

- CS0 – Sennheiser ME 104 (blízká vzdálenost)
- CS1 – Nokia Lavalier HDC-6D (blízká vzdálenost)
- CS2 – Sennheiser ME 64 (střední vzdálenost)
- CS3 – Haun MBNM-550 E-L (velká vzdálenost)

Mikrofon v kanále CS0 má oproti stejně vzdálenému mikrofonu CS1 vyšší citlivost a nižší úroveň vlastního šumu. Oba tyto mikrofony byly při pořízení nahrávek umístěny na náhlavní soupravě.

Střední a velká vzdálenost mikrofonu se liší podle místa nahrávek. U skupiny mluvčích OFFICE a ENTERTAINMENT byl mikrofon CS3 umístěn přibližně ve vzdálenosti 3 metry a CS2 ve vzdálenosti 1 metr od mluvčího. U nahrávek pořízených v automobilu ale byly mikrofony CS2 a CS3 umístěny v podstatně menší vzdálenosti, jelikož musely být umístěny v interiéru vozu.

Vliv na chybu identifikace a verifikace pro výchozí nastavení parametrů u všech 4 kanálů ukazuje tab. 19. Dále byly samostatně otestovány skupiny OFFICE (tab. 20) a PUBLIC PLACE (tab. 21) na identifikaci v uzavřené množině. U ostatních skupin je pro rozumné vyhodnocení chyby málo mluvčích. V kanceláři byl mikrofon CS3 umístěn ve vzdálenosti 3 metry a na veřejném prostranství 1 metr. Vyšší chybu CS3 (v kanceláři) kromě větší vzdálenosti mikrofonu od zdroje způsobil pravděpodobně ještě dozvuk.

Kanál	IVEC P_{err} [%]	PLDA P_{err} [%]
CS0	0	0
CS1	0	0
CS2	0,70	0,35
CS3	1,76	1,76
Kanál	IVEC EER [%]	PLDA EER [%]
CS0	1,40	0
CS1	3,52	1,05
CS2	4,61	2,82
CS3	10,04	5,01

Tab. 19 Chybovost (SPEECON – NUMERIC+SENTENCE) pro různé kanály nahrávek, 284/568 mluvčích

Kanál	IVEC P_{err} [%]	PLDA P_{err} [%]
CS0	0	0,47
CS1	0,95	0
CS2	0,47	0
CS3	3,80	3,80

Tab. 20 Chyba identifikace (SPEECON – NUMERIC+SENTENCE) pro různé kanály nahrávek (OFFICE), 210/210 mluvčích

Kanál	IVEC P_{err} [%]	PLDA P_{err} [%]
CS0	0	0
CS1	0,47	0
CS2	1,42	1,90
CS3	2,38	1,42

Tab. 21 Chyba identifikace (SPEECON – NUMERIC+SENTENCE) pro různé kanály nahrávek (PUBLIC PLACE), 210/210 mluvčích

Pro otestování metod LDA a PLDA u databáze SPEECON byl pro jednotlivé nahrávky od daného mluvčího střídán (alternován) kanál nahrávek (typ mikrofonu) CS0 až CS3. Oproti standardním parametrům bylo dosaženo při testu na počet GMM komponent lepších výsledků při 128 komponentách namísto výchozích 64. Při porovnání s jinými experimenty je zde PLDA metoda výrazně efektivnější, jelikož umožňuje dobře rozlišit konvoluční šum různých mikrofonů. Test metodou PLDA vykazuje 4krát menší chybu identifikace nežli základní test IVEC, viz tab. 22.

Nalezení optimálních hodnot parametrů `LDA_DIM` a `LDA_COVAR_FACTOR` (viz implementace 4.2.5) bylo provedeno, viz tab. 23, přičemž chyba před LDA transformací byla 2,11 %, viz tab. 22. U experimentů, u kterých dimenze po LDA transformaci nebyla nastavena na optimální hodnotu, bylo možné v tomto rozsahu najít minimální chybu identifikace u hodnot kovariančního faktoru mezi 0,02 až 0,1.

IVEC P_{err} [%]	LDA P_{err} [%]	PLDA P_{err} [%]
2,11	1,76	0,35

Tab. 22 Alternující kanál nahrávek (SPEECON – NUMERIC+SENTENCE). Byly použity různé typy mikrofonů pro trénovací i testovací data, 284/568 mluvčích.

Dimenze LDA	LDA P_{err} [%]	Kovarianční faktor	LDA P_{err} [%]
400	2,11	0,500	3,52
350	2,46	0,200	2,46
300	2,46	0,100	1,76
250	**1,76**	0,050	1,76
200	2,11	0,020	1,76
150	4,22	0,010	1,76
100	4,57	0,005	1,76
50	4,92	0,002	1,76

Tab. 23 Hledání nejmenší chyby identifikace (SPEECON – NUMERIC+SENTENCE) po LDA transformaci pomocí dvou nastavitelných parametrů během generování transformační matice $\boldsymbol{A}$, 284/568 mluvčích.

5.2.7 Vliv délky rozpoznávaných promluv

Pro výchozí hodnoty trénovací množiny dat NUMERIC+SENTENCE byla otestována chybovost pro malou testovací množinu promluv složenou z několika málo samostatných slov (resp. vět). Do výběru rozpoznávaných (testovacích) nahrávek u databáze SPEECON byly od každého mluvčího zahrnuty 4 promluvy W01–04 (resp. S01–04) namísto standardních promluv CC4 a S04. Analogicky byly u databáze SpeechDat nahrazeny promluvy C4 a S4 množinou promluv W0–4 (resp. S0–4). Místo pěti číslovek a jedné věty byla v testovací množině nahrávek od každého mluvčího k dispozici maximálně 4 slova (resp. věty).

UBM model a extraktor i-vektorů byl natrénován pouze jednou, a v jednotlivých průchodech fáze STAGE 0–1 a STAGE 3–5 byl snižován počet slov (resp. vět) v testovací množině promluv, viz tabulky 24 a 25 pro slova a tabulky 26 a 27 pro věty. Z uvedených výsledků vyplývá, že přesnost popisu každého mluvčího reprezentovaného i-vektory řádově v počtu jednotek slov je velmi citlivá na změnu množství dat. U experimentů v tabulkách 24 a 25 není vynesena chyba verifikace, jelikož pro vysoké chyby P_{err} nemá již EER příliš vypovídající hodnotu.

Promluvy	IVEC P_{err} [%]	PLDA P_{err} [%]
W01–4	6,69	0,70
W01–3	12,67	0,70
W01–2	15,49	2,11
W01	25,35	13,73

Tab. 24 Chyba identifikace (SPEECON – NUMERIC+SENTENCE) pro různé množství testovacích promluv (slov), 284/568 mluvčích

Promluvy	IVEC P_{err} [%]	PLDA P_{err} [%]
W1–4	14,82	2,66
W1–3	16,92	4,37
W1–2	22,05	7,03
W1	34,22	20,53

Tab. 25 Chyba identifikace (SpeechDat – NUMERIC+SENTENCE) pro různé množství testovacích promluv (slov), 526/1052 mluvčích

Promluvy	IVEC P_{err} [%]	PLDA P_{err} [%]
S04–7	0	0
S04–6	0	0
S04–5	0,35	0,35
S04	1,76	0,70
Promluvy	IVEC EER [%]	PLDA EER [%]
S04–7	0,70	0,35
S04–6	0,70	0,35
S04–5	1,76	1,06
S04	6,09	2,48

Tab. 26 Chybovost (SPEECON – NUMERIC+SENTENCE) pro různé množství testovacích promluv (vět), 284/568 mluvčích

Promluvy	IVEC P_{err} [%]	PLDA P_{err} [%]
S4–7	0	0
S4–6	0,19	0
S4–5	0,19	0,19
S4	2,47	0,57
Promluvy	IVEC EER [%]	PLDA EER [%]
S4–7	3,04	1,14
S4–6	3,23	1,33
S4–5	5,52	1,90
S4	7,79	4,01

Tab. 27 Chybovost (SpeechDat – NUMERIC+SENTENCE) pro různé množství testovacích promluv (vět), 526/1052 mluvčích

5.2.8 Optimalizace prahu u verifikační fáze

Pro výchozí parametry dle tab. 2 bylo otestováno okolí verifikačního prahu chybové míry EER. Nejlepších výsledků bylo dosaženo u obou databází metodou PLDA. Chybové míry P_{FA} a P_{miss} byly vypočteny dle vztahu (4) a (5) a vyneseny do tabulek 28 a 29.

Pro relativně nízký počet mluvčích a malou chybovost u PLDA by nebylo zobrazení DET křivky přehledné. Jelikož se hodnoty P_{FA} a P_{miss} téměř neprotínají, pro zobrazení DET křivky je k dispozici velmi nízké rozlišení. U databáze SpeechDat bez použití PLDA už lze DET křivku přehledně vynést do grafu (obr. 13). Parametry DCF funkce byly převzaty z evaluačního plánu pro rozpoznávání mluvčího NIST z roku 2008 [3]. Hodnoty parametrů jsou: $C_{miss} = 10$, $C_{FA} = 1$, $P_{target} = 0,01$. Pro tyto vstupní parametry dle vztahu (8) je C_{det}^{min} rovna hodnotě 0,0293 a optimální chybové míry jsou $P_{FA} = 1{,}71$ % a $P_{miss} = 12{,}38$ %.

Práh (PLDA)	P_{FA} [%]	P_{miss} [%]
-30	5,28	0
-25	2,81	0
-20	1,40	0
-15	0,70	0
-10	**0**	**0,35**
-5	0	1,05
0	0	2,81
5	0	7,39

Tab. 28 FAR(P_{FA}) a FRR(P_{miss}) (SPEECON – NUMERIC+SENTENCE) pro různé hodnoty zvoleného prahu PLDA skóre, 284/568 mluvčích. Hodnota EER = 0 %.

Práh (PLDA)	P_{FA} [%]	P_{miss} [%]
-15	7,79	0
-10	4,18	0,38
-5	3,04	0,57
0	**1,33**	**0,95**
5	1,33	1,90
10	0,95	4,56
15	0,76	8,55

Tab. 29 FAR(P_{FA}) a FRR(P_{miss}) (SpeechDat – NUMERIC+SENTENCE) pro různé hodnoty zvoleného prahu PLDA skóre, 526/1052 mluvčích. Hodnota EER = 1,33 %.

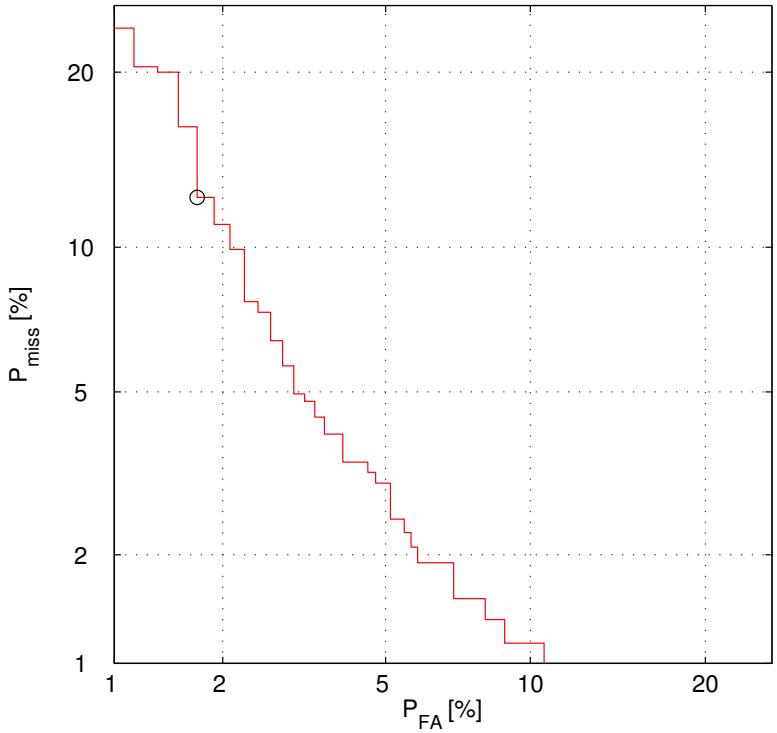
5.3 Srovnání výsledků s jinými experimenty

Nejlepších výsledků bylo dosaženo pro trénovací promluvy dlouhé 30 sekund a testovací promluvy dlouhé 10 sekund (NUMERIC+SENTENCE) od každého mluvčího, viz tab. 30. Z databáze SPEECON bylo použito 568 mluvčích a u databáze SpeechDat všech 1 052 mluvčích. V obou případech bylo UBM natrénováno s 64 GMM komponentami. Nulová hodnota znamená, že se podařilo identifikovat a verifikovat všechny mluvčí z referenční množiny, kterých je v tomto případě polovina z celkového počtu mluvčích. Skutečná chyba systému se pohybuje mezi 0–0,18 % pro SPEECON a 0–0,095 % pro SpeechDat.

Databáze	PLDA P_{err} [%]	PLDA EER [%]
SPEECON	0	0
SpeechDat	0	1,33

Tab. 30 Nejlepší výsledky pro výchozí nastavení parametrů úlohy v1 při použití množiny nahrávek NUMERIC+SENTENCE a poloviční počtu referenčních mluvčích

Všechny výpočty uvedené v tabulkách byly provedeny na SGE (HW konfigurace v podkapitole 5.2.1), ale pro lepší představu o náročnosti úlohy v1 na CPU uvedu reálné časy výpočtů na PC, který byl používán při ladění úlohy. Výpočet byl proveden na 2jádrovém procesoru typu Intel Core 2 Duo s taktovací frekvencí 2,4 GHz. Reálné časy výpočtů některých fází jsou uvedeny v tabulce 31. Pokud bude například ověřován krátký úsek řeči dlouhý 10 sekund, bude nutné provést výpočet jeho charakteristik a extrahovat pro něj i-vektory. Pokud bychom zanedbali režijní čas pro spuštění



Obr. 13 DET křivka pro IVEC skóre (SpeechDat – NUMERIC+SENTENCE), 526/1052 mluvčích. DCF vypočítána dle parametrů NIST 2008. Chyba EER = 4 % a P_{FA} = 1,71 %.

těchto výpočtů, i-vektor reprezentující tento úsek řeči by měl být připraven za přibližně 0,7 sekundy (SpeechDat).

Databáze	MFCC + VAD	Trénování modelu	Extrakce i-vektorů
SPEECON	3,5	09,2	4,5
SpeechDat	4,5	14,5	7,8

Tab. 31 Reálný čas výpočtů (PC) v minutách u fáze STAGE 1 až STAGE 3. Jedná se o výchozí nastavení parametrů úlohy v1 při použití množiny nahrávek NUMERIC+SENTENCE a polovičním počtu referenčních mluvčích.

Pro srovnání uvádím výsledky úlohy s podobnými vstupními podmínkami. Autor [1] prezentuje ve své práci chybu identifikace v uzavřené množině P_{err} = 0,4 %. Jednalo se o GMM model, který byl natrénován 10 komponentami a databáze obsahovala 500 mluvčích s přibližně stejným poměrem mužů a žen. Použité nahrávky byly v telefonní kvalitě μ-law. Od každého mluvčího bylo k dispozici několik krátkých vět a izolovaných slov. Stejně jako v mém případě byly pro charakteristiku mluvčího použity MFCC koeficienty dimenze 20. Nutno podotknout, že tento systém ještě nepoužíval reprezentaci mluvčích jako i-vektorů ani metody pro zpřesnění klasifikace jako LDA a PLDA.

Další srovnání mohu provést s výsledky NIST evaluací z roku 2008 a 2010 (viz [3]). Zde jsou používány GMM modely s 512 až 2 048 komponentami. Omezení délky trénovací i testovací množiny dat je pro všechny účastníky evaluací 5 minut telefonního hovoru ve formátu μ-law. Nahrávky jsou dvoukanálové, tedy každá strana je nahrávána do vlastního kanálu. V našem případě vezmeme v úvahu podmínku „tel – tel“,

kdy všichni mluvčí mluvili stejným jazykem (anglickým). Pro i-vektorový systém bylo dosaženo následujících výsledků. V roce 2008 bylo natrénováno celkem 3 263 mluvčích (modelů) a bylo uskutečněno 98 776 verifikačních soudů. Pro ženské verifikační soudy bylo dosaženo EER = 5,58 % a pro mužské EER = 6,38 %. V roce 2010 už bylo natrénováno 5 460 mluvčích (nezávisle na pohlaví) a provedeno 610 748 soudů, poté bylo dosaženo chyby EER = 5,78 %. V mém případě se u databáze SpeechDat jednalo pouze o 1 052 soudů a z toho referenčních mluvčích byla polovina, poté bylo EER = 1,33 %.

6 Závěr

V rámci této práce jsem prostudoval moderní metody identifikace mluvčího a zaměřil jsem se především na ty nejrozšířenější, jež modelují mluvčího statisticky, na bázi i-vektorů. Dále jsem v práci věnoval pozornost metodám pro zvýšení přesnosti klasifikace v prostoru i-vektorů (LDA a PLDA), které lze použít volitelně. Za účelem výběru vhodných příznaků řeči pro využití v úloze identifikace byly zjištěny vlastnosti a přesnost nejpoužívanějších příznaků (viz 2.2.3). Jako nejvhodnější příznaky se pro tuto úlohu jeví mel-frekvenční kepstrální koeficienty (MFCC), jež jsou popsány v podkapitole 2.2.2.

Pro implementaci byla zvolena sada nástrojů KALDI, s jejíž pomocí jsem implementoval skripty („recepty“) za účelem otestování úlohy identifikace a verifikace mluvčího na dvou vybraných databázích. Pro tyto účely byla vybrána nejprve databáze širokopásmových nahrávek SPEECON, poté databáze nahrávek v telefonní kvalitě SpeechDat. Implementace úlohy bude dostupná odborné komunitě v oblasti rozpoznávání mluvčího.

V experimentální části bylo odladěno nastavení úlohy na obou databázích pro takové množství dat (viz 5.2), které má simulovat reálné nasazení aplikace v praxi. Toto nastavení bylo stanoveno jako výchozí pro další série experimentů, u kterých byly postupně zhoršovány jednotlivé parametry nebo vstupní podmínky, za účelem zjištění hraničních hodnot použitelnosti. V každém experimentu byla vyhodnocena chyba identifikace mluvčího a u většiny experimentů následně i chyba verifikace.

Byly otestovány důležité parametry modelu, a to počet GMM komponent a dimenze i-vektorů. Další experimenty se týkaly šířky pásma nahrávek, a to jednak snížením vzorkovací frekvence širokopásmových dat, ale také porovnáním přesnosti mezi databázemi SPEECON a SpeechDat na přibližně stejném množství dat a stejném počtu mluvčích. Dále byl ověřen vliv velikosti dimenze příznakového vektoru MFCC pro dvě různě velké množiny dat. Pro různé skupiny (místa nahrávek) mluvčích v rámci každé databáze byla vyhodnocena identifikace v uzavřené množině. U databáze SPEECON byl otestován vliv použitého typu mikrofonu a jeho vzdálenosti při pořízení nahrávek na přesnost identifikace, poté bylo otestováno chování metod LDA a PLDA při alternaci těchto mikrofonů v rámci jednoho experimentu. Dále byla ověřena přesnost identifikace pro různé množství rozpoznávaných promluv. Nakonec byla pro výchozí nastavení vyhodnocena verifikační fáze úlohy pomocí změny prahu a DET křivky.

V jednotlivých sériích experimentů byly potvrzeny očekávané výsledky experimentů u obou databází a v závěru experimentální části bylo provedeno srovnání výsledků s jinými projekty v úloze rozpoznávání mluvčího, viz 5.3.

U provedených experimentů (viz 5.3) se ve výchozím nastavení podařilo pomocí metody PLDA u databáze SPEECON bezchybně identifikovat a následně verifikovat všech 284 referenčních mluvčích. Z množiny všech 590 mluvčích databáze bylo pro účely rozpoznávání nicméně vyřazeno 22 mluvčích, u kterých byly nahrávky silně zarušeny. U databáze SpeechDat bylo za podobných podmínek pro všechny mluvčí dosaženo chyby verifikace EER = 1,33 %, zajímavé je ale srovnání, že při použití všech nahrávek databáze byla chyba EER pouze 0,76 % (viz tab. 6).

Příloha A

Obsah přiloženého CD

Přiložené CD obsahuje soubory popsané v následujících bodech.

- Text diplomové práce ve formátu PDF (`Moderni_metody_rozpoznavani_mluvciho_na_bazi_GMM_a_DNN.pdf`).
- Implementaci úlohy komprimovanou ve formátu ZIP (`Speaker_recognition_in_KALDI.zip`). Po rozbalení souboru implementace je k dispozici úloha identifikace mluvčího pro každou databázi (`speecon` a `speechdat`) ve vlastním adresáři.

Bibliografie

1. ŠVEDNA, Z. *Rozpoznávání řečníka*. 2003. Disertační práce. Západočeská univerzita v Plzni.
2. PSUTKA, J.; MÜLLER, L.; MATOUŠEK, J.; RADOVÁ, V. *Mluvíme s počítačem česky*. Praha: Academia, 2006. ISBN 80-200-1309-1.
3. SILOVSKÝ, J. *Generativní a diskriminativní klasifikátory v úlohách textově nezávislého rozpoznávání a diarizace mluvčích*. 2011. Disertační práce. Technická univerzita v Liberci.
4. DODDINGTON, G. *DET-Curve Plotting software for use with MATLAB* [online] [visited on 2016-07-20]. Available from: `http://www.itl.nist.gov/iad/mig/tools/DETware_v2.1.targz.htm`.
5. KOCKMANN, M. *Modelování prozodických příznaků pro ověřování mluvčího v podprostorech*. 2011. Disertační práce. Vysoké učení technické v Brně.
6. MONGWE, W. *Understanding the EM Algorithm* [online]. 2015 [visited on 2016-05-13]. Available from: `http://www.wilsonmongwe.co.za/understanding-the-em-algorithm`.
7. AVILA, A. R.; MILTON, S. P.; FRAGA, F. J.; O'SHAUGHNESSY, D.; FALK, T. H. Improving the Performance of Far-Field Speaker Verification Using Multi-Condition Training: The Case of GMM-UBM and i-vector Systems. In: *Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association*. Singapore, 2014.
8. MATĚJKA, P.; GLEMBEK, O.; CASTALDO, F.; ALAM, M.J.; PLCHOT, O.; KENNY, P.; BURGET, L.; CERNOCKY, J. Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification. In: *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Praha, 2011.
9. POVEY, D. et al. *The Kaldi Speech Recognition Toolkit* [online] [visited on 2016-09-28]. Available from: `http://kaldi-asr.org/doc/`.
10. MACHLICA, L. *Vysokodimenzionální prostory a modelování v úloze rozpoznávání řečníka*. 2012. Disertační práce. Západočeská univerzita v Plzni.
11. RICHARDSON, F.; REYNOLDS, D.; DEHAK, N. Deep Neural Network Approaches to Speaker and Language Recognition. *IEEE Signal Processing Letters*. 2015, vol. 22, no. 10.
12. MCLAREN, M.; LEI, Y.; FERRER, L. Advances in Deep Neural Network Approaches to Speaker Recognition. In: *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Brisbane, Australia, 2015.
13. PREZIOSO, M.; MERRIKH-BAYAT, F.; HOSKINS, B.; ADAM, G.; LIKHAREV, K.; STRUKOV, D. *Training and operation of an integrated neuromorphic network based on metal-oxide memristors* [online]. 2015 [visited on 2016-09-12]. Available from: `http://dx.doi.org/10.1038/nature14441`.

14. WOODLAND, P. *The Hidden Markov Model Toolkit (HTK)* [online] [visited on 2016-09-15]. Available from: `http://htk.eng.cam.ac.uk/`.

15. POVEY, D. et al. The Kaldi Speech Recognition Toolkit. In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Hilton Waikoloa Village, Big Island, Hawaii, US, 2011.

16. POLLÁK, P.; ČERNOCKÝ, J. *Czech SPEECON Adult Database*. 2004. Dokumentace databáze, ČVUT Praha a VUT Brno.

17. POLLÁK, P.; ČERNOCKÝ, J.; HANŽL, V. *Czech SpeechDat(E) Database*. 2000. Dokumentace databáze, ČVUT Praha a VUT Brno.