Fakulta aplikovaných věd

katedra kybernetiky

Disertační práce
k získání akademického titulu doktor
v oboru Kybernetika

Ing. Zbyněk Zajíc

Adaptace akustického modelu v úloze s malým množstvím adaptačních dat

školitel: Doc. Dr. Ing. Vlasta Radová

Plzeň, 2012

FACULTY
OF APPLIED SCIENCES
UNIVERSITY
OF WEST BOHEMIA

Faculty of Applied Sciences

Department of Cybernetics

Doctoral thesis
submitted for the degree Doctor of Philosophy
in the field of Cybernetics

Ing. Zbyněk Zajíc

Adaptation of an Acoustic Model in the Task of Small Amount of Adaptation Data

Advisor: Doc. Dr. Ing. Vlasta Radová

Pilsen, 2012

Prohlášení

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

V Plzni dne Zbyněk Zajíc

Poděkování

Tato dizertační práce vznikla za odborného vedení mé školitelky Doc. Dr. Ing. Vlasty Radové. Dále bych chtěl poděkovat za odborné rady a konzultace Doc. Ing. Luďkovi Müllerovi, Ph.D., vedoucímu oddělení umělé inteligence na katedře kybernetiky.

Dále bych chtěl poděkovat své rodině za vytvoření dobrých pracovních podmínek a kolegům z oddělení umělé inteligence katedry kybernetiky za cenné rady a pomoc při vypracovávání této práce.

Tato práce vznikla za finanční podpory projektu "Eliminace jazykových bariér handicapovaných diváků České televize"(MŠMT 2C06020) a s možností využití výpočetních prostředků MetaCentra VO poskytovaných programem "Velká infrastruktura CESNET"(LM2010005).

Obsah

Seznam zkratek

Acc	Accuracy
ANN	Artifcial Neural Networks
ASR	Automatic Speech Recognition
BIC	Bayes Information Criterion
BW	Baum-Welch
CAT	Cluster Adaptive Training
CF	Certainty Factor
CLT	Central Limit Theorem
CMLLR	Constrained Maximum Likelihood Linear Regression
CMN	Cepstrum Mean Normalization
Corr	Correctness
CzT	Český telefonní korpus
D	Delete
DAT	Discriminative Adaptation Training
DCAT	Discriminative CAT
DCT	Discrete Cosine Transformation
DfMLLR	Discriminative fMLLR
DLLR	Discounted Likelihood Linear Regression
DLT	Discriminative Linear Transformation
DMAP	Discriminative MAP
DMLLR	Discriminative MLLR
DT	Discriminative Training
DTW	Dynamic Time Warping
EBW	extended Baum-Welch
ED	Eigenvoices Decomposition
EF	Eigen Face
EM	Expectation-Maximization
EV	Eigen Voices
FA	Factor Analysis
fMLLR	feature Maximum Likelihood Linear Regression
GD	Gender Dependent
GMM	Gaussian Mixture Model
H	Hit
HMM	Hidden Markov Model
I	Inzertion
ICA	Independent Component Analysis
iid	independent and identically distributed
IRPROP	Improved Resilient Propagation
JFA	Joint Factor Analysis

KEV	Kernel Eigen Voices
L-BFGS	Limited memory Broyden, Fletcher, Goldfarb and Shanno
LD	Linear Discriminant
LLR	Log Likelihood Ratio
LM	Language Model
LSE	Least Square Error
LT	Linear Transformation
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A-Posteriori Probability
MAPLR	Maximum A-Posterior Probability Linear Regression
MCE	Minimum Classification Error
MFCC	Mel Frequency Cepstral Coefficient
ML	Maximum Likelihood
MLED	Maximal Likelihood Eigenvoices Decomposition
MLLR	Maximum Likelihood Linear Regression
MLLRcov	MLLR for covariance matrix
MLLRmean	MLLR for mean
MLP-ANN	Multi-layer Perceptron ANN
MMI	Maximum Mutual Information
MMI-FD	Maximum Mutual Information Frame Discrimination
MPE	Minimum Phone Error
MWE	Minimum Word Error
OOV	Out Of Vocabulary
PCA	Principal Component Analysis
PCMLLR	Predictive CMLLR
PLP	Perceptual Linear Predictive
RMP	Regression-based Model Prediction
RT	Regresion Tree
S	Substitution
SA	Speaker Adaptive
SAT	Speaker Adaptive Training
SC	Speaker Clustering
SCD	Speaker Change Detection
SD	Speaker Dependent
SD-E	SpeechDat-East korpus
SI	Speaker Independent
SLAPT	Sine-log all-pass transformation
SMAP	Structural Maximum A Posteriori
SMAPLR	Structural Maximum A Posteriori Linear Regression
SV	Speaker Verification
SVD	Singular Value Decomposition
UBM	Universal Background Model
VAD	Voice Activity Detector
VFS	Vector Field Smoothing
VTLN	Vocal Tract Length Normalization
VTLN-LT	VTLN Linear Transformation
WER	Word Error Rate
WNR	Weighted Neighbor Regression
WSMAP	Weighted Structural Maximum A Posteriori
(f)MLLR	fMLLR, MLLR

Seznam tabulek

Seznam obrázků

Anotace

Tato práce se zabývá problematikou automatické adaptace akustického modelu na aktuální data od konkrétního řečníka. Pro natrénování modelu je potřeba velkého množství dat, které je z praktického hlediska nemožné získat od jednoho řečníka. Řešením je konstrukce akustického modelu na datech od více řečníků a následná adaptace tohoto modelu na dostupných datech daného řečníka. Klasické metody adaptace, představené v této práci, mají problémy s malým množstvím adaptačních dat, takto adaptovaný model může ve výsledku zhoršovat rozpoznávání.

Práce si klade za cíl vysvětlit principy používaných adaptačních metod a postupy adaptačního trénování, dále se zaměřuje na problém nedostatku dat při adaptaci. Jsou zde představeny známé robustní metody adaptace a navržena vlastní řešení, jejichž účinnost je vzájemně experimentálně porovnána.

Anotation

This work is focused on the automatic speaker adaptation of an acoustic model, which is a part of the automatic speech recognition system. To train the acoustic model it is necessary to have large amount of data from many speakers. The final speaker-independent model is then able to recognize the speech from any speaker. The speaker-independent model is adapted to the speech of a specific speaker. Ordinary adaptation techniques introduced in this work perform poorly in cases with insufficient amount of adaptation data. The aim of this work is to discuss methods of adaptation and adaptation training. To avoid the problem with lack of adaptation data various robust solutions have been described and new one have been proposed. Some of these methods were tested, and the experiments show that the robust adaptation contributes significantly to the task of automatic speech recognition.

Kapitola 1

Úvod

Řeč, jako jeden z nejpoužívanějších způsobů předávání informací mezi lidmi, je v popředí zájmu oboru umělé inteligence již několik desítek let. Mezi problémy zpracování řeči počítačem patří, mimo jiné, úloha **automatického rozpoznávání řeči** (ASR – Automatic Speech Recognition), tedy úloha přepisu mluveného slova na text pomocí stroje. První automatické rozpoznávače se objevily v šedesátých letech minulého století, avšak jejich úspěšnost byla značně omezena tehdejšími možnostmi výpočetní techniky. První rozpoznávače se soustředily pouze na přepis izolovaných slov. Teprve v sedmdesátých letech, s příchodem myšlenky **skrytých Markovových modelů** (HMM – Hidden Markov Model) a prudkým rozvojem výpočetní techniky, došlo k nastartování vývoje systémů ASR a jejich směřování k rozpoznávání řeči spojité.

Se zdokonalováním ASR začal také růst počet slov obsažených v rozpoznávacím slovníku, z několika stovek v osmdesátých letech na několik tisíc v letech devadesátých. Systémy využívající slovník s velkým počtem slov se odborně označují **LVSCR** (Large Vocabulary Continuous Speech Recognition) systémy [1]. Také kvalita rozpoznávané řeči přešla z čistých laboratorních dat k spontánním hovorům v rušném prostředí.

V současné době, kdy je obvyklé rozpoznávat spontánní hovory ve špatné akustické kvalitě, čelíme mnoha problémům [2]. Jedním z nich jsou právě různé akustické podmínky v nahraných datech způsobené rozdílným nahrávacím prostředím, různým kanálem a odlišným řečníkem. To vše přidává nežádoucí varianci v nahraných datech. Při rozpoznávání testovacích dat s jinými akustickými vlastnostmi, než měla trénovací data použitá pro vytvoření akustického modelu, dochází k degradaci úspěšnosti rozpoznávání. Řešením by bylo použít model natrénovaný na datech se stejnými akustickými podmínkami jako v testovaných datech, to však v principu není zcela možné. Například získání dostatečného množství dat od jednoho řečníka pro natrénování akustického modelu je v praxi nereálné.

Z tohoto důvodu jsou již dvacet let vyvíjeny **adaptační techniky** normalizující testovací data nebo posouvající parametry akustického modelu směrem k testovacím datům. Úspěšnost rozpoznávání může být díky adaptaci výrazně zlepšena, a to již při použití několika málo promluv od cílového řečníka. Zároveň s řečníkem jsou adaptovány i akustické podmínky při nahrávaní, jako jsou typické ruchy prostředí, použité nahrávací zařízení atd.

Úkolem trénování je vytvořit model dobře odpovídající testovaným datům. V praxi však obecně máme nehomogenní data, která obsahují směs různých akustických zdrojů. Natrénovaný model se pak nazývá **multi-style model**. Tento model je možno použít pro testování nebo jej dále adaptovat na testované akustické podmínky, čímž se zvýší jeho efektivita pro testovaná data. Problém velké variability v trénovacích datech tím ale není úplně odstraněn. Řešením je **adaptační trénování**, jehož úkolem je snížit variabilitu z trénovacích dat a vytvořit tzv. **kanonický model**, z něhož je vyloučena jakákoliv informace o prostředí či řečníkovi.

Kanonický model je následně adaptován na testovací podmínky.

S novým využitím ASR v **on-line** aplikacích [3] vyvstaly nové problémy pro adaptaci, které zahrnují vyřešení specifických úkolů souvisejících s on-line zpracováním mluvené řeči. Při on-line rozpoznávání neznáme dopředu identitu rozpoznávaného řečníka, tedy adaptace musí proběhnout až v průběhu rozpoznávacího procesu na aktuálně rozpoznávaných datech. Těchto dat je obvykle velmi malé množství, což kontrastuje s požadavkem rychlé adaptace na řečníka. Kromě toho data pro on-line adaptaci nemají referenční přepis. Proto byly vyvinuty metody **robustní adaptace**, které se snaží předcházet problémům s malým množstvím nepřesně přepsaných dat pro adaptaci.

Tato práce si klade za cíl vysvětlit principy používaných adaptačních metod a postupy adaptačního trénování. Adaptace je zde popisována jako přizpůsobení se cílovému řečníku, ale z principu věci jde vlastně o obecnou adaptaci na akustické podmínky, protože cílový řečník není nic jiného než jiný akustický kanál pro přenos hlasu. Dále je práce zaměřena na robustní přístupy k adaptaci, převážně se snaží řešit problém malého množství dat pro adaptaci. Jsou zde popsány používané robustní přístupy spolu s navrženým vlastním řešením. Tyto postupy jsou experimentálně ověřeny.

1.1 Cíle disertační práce

- Popsat principy nejpoužívanějších metod adaptace vycházející jak z generativního tak i z diskriminativního přístupu.
- Prozkoumat existující robustní přístupy k adaptaci zaměřující se na problém malého množství dat bez referenčního přepisu.
- Zaměřit se na zlepšení účinnosti metod adaptace, převážně pak metod založených na lineárních transformacích, které vykazují dobré vlastnosti i pro malý počet adaptačních dat.
- Provést experimentální porovnání jednotlivých metod, převážně pak robustních přístupů (ať již převzatých nebo vlastních) s důrazem na jejich účinnost s malým množstvím adaptačních dat.
- Implementovat robustní přístupy adaptace do on-line systému pro rozpoznávání mluveného slova.

1.2 Struktura práce

Předložená práce se zabývá adaptací akustického modelu, proto je v následující kapitole nejprve popsán akustický model a postupy pro jeho natrénování. Dále je v kapitolách 3 a 4 uveden současný stav z hlediska různých přístupů k adaptaci.

V kapitole 3 je pozornost věnována přístupům založeným na adaptaci akustického modelu. Uvedeny jsou zde jednak metody lineární transformace (LT), které v současné době patří k nejpoužívanějším, a jednak metoda maximalizace aposteriorní pravděpodobnosti (MAP). Další část kapitoly 3 je pak věnována kombinacím těchto metod a jiným odvozeným přístupům.

Kapitola 4 se zabývá jiným přístupem blízkým k adaptaci, a to tzv. adaptačním trénováním, při kterém je odstraňována nežádoucí variability v akustickém modelu, a tím usnadněna jeho následná adaptace na konkrétního řečníka. Adaptační trénování využívá postupy odvozené z adaptace, ale aplikuje je na trénovací data.

Problémům on-line adaptace se věnuje kapitola 5, kde je stručně zmíněn i největší z problémů on-line přístupu, a tím je nedostatek dat pro adaptaci. Tomuto problému je pak věnována celá následující kapitola 6, která popisuje existující řešení nedostatku adaptačních dat.

V kapitole 7 jsou popsány srovnávací experimenty jednotlivých nejpoužívanějších adaptačních metod a adaptačního trénování. Dále jsou zde uvedeny vlastní návrhy pro zlepšení adaptace v úloze malého počtu adaptačních dat, které jsou experimentálně porovnány s existujícími postupy. Pozornost je věnována i experimentům zaměřeným na rozdílné množství adaptačních dat.

Závěr práce, kapitola 8, poté shrnuje dosažené výsledky.

Kapitola 2

Akustické modelování

Tato kapitola si klade za cíl přiblížit čtenáři základní principy modelování řeči pomocí akustického modelu reprezentovaného **skrytými Markovovými modely** (HMM – Hidden Markov Model). Je zde popsána struktura modelu a postupy při rozpoznávání posloupnosti řeči. Hlavní důraz je kladen na metody konstrukce HMM, neboť ty jsou základem adaptačních technik, jimiž se tato práce zabývá. Detailní popis trénování i využití skrytého Markovova modelu je možno nalézt v [4], [5] nebo [6].

2.1 Struktura akustického modelu

Při rozpoznávání souvislé řeči jsou v dnešní době nejvíce dominantní klasifikátory pracující se statistickými metodami, kdy jsou slova (častěji subslovní jednotky jako slabiky, fonémy, trifóny a pod.) modelovány pomocí HMM. Vyslovená posloupnost slov W je nejprve rozčleněna na krátkodobé úseky, tzv. mikrosegmenty, po jejichž dobu předpokládáme, že parametry hlasového ústrojí jsou stacionární. Pro každý mikrosegment je vypočítán vektor příznaků $\boldsymbol{o}(t)$, který tvoří parametrizovaný přepis vyřčené promluvy $\boldsymbol{O} = \{\boldsymbol{o}(1), \boldsymbol{o}(2), \ldots, \boldsymbol{o}(T)\}$. Celá promluva je modelována zřetězením subslovních modelů HMM sériově za sebou. Cílem rozpoznávání je pak nalézt posloupnost slov W^*, která maximalizuje podmíněnou pravděpodobnost $P(\boldsymbol{O}|W)$ pro danou akustickou informaci $\boldsymbol{O}$.

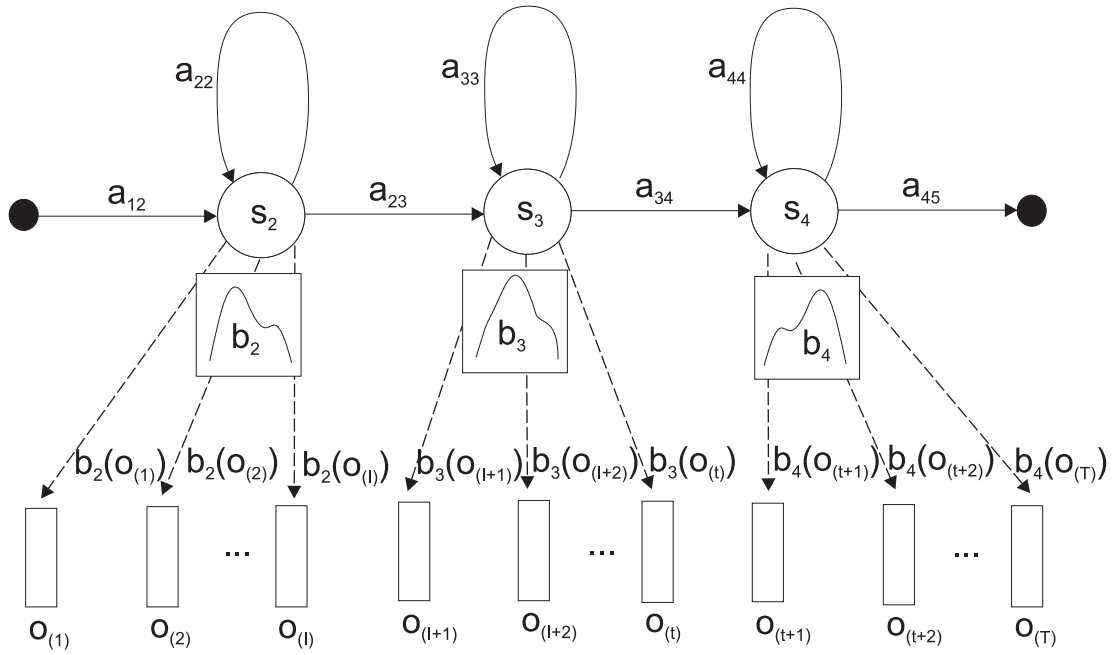
Jako akustický model je uvažován skrytý Markovův model, patřící do množiny pravděpodobnostních konečných automatů, které mají tzv. Markovovu vlastnost, tedy současný stav modelu je závislý pouze na n stavech předcházejících. Skrytý se nazývá proto, že pozorovatel vidí jen výstup, ale posloupnost stavů modelu je mu skryta. Používají se zejména tzv. levo-pravé Markovovy modely, které jsou zvláště vhodné pro modelování procesů jako je spojitá řeč, jejichž vývoj je spojen s postupujícím časem.

Na skrytý Markovův model (příklad na obrázku 2.1) lze pohlížet jako na pravděpodobnostní konečný automat, který přechází z jednoho stavu do stavu druhého přes předem dané pravděpodobnostní přechody a tím generuje náhodnou posloupnost pozorování $\boldsymbol{O} = \{\boldsymbol{o}(1), \boldsymbol{o}(2), \ldots, \boldsymbol{o}(T)\}$. Stav s_j, do kterého model přejde, generuje příznakový vektor $\boldsymbol{o}(t)$ podle rozdělení výstupní pravděpodobnosti $b_j(\boldsymbol{o}(t))$.

Podmíněná pravděpodobnost přechodu a_{ij} určuje, s jakou pravděpodobností přechází model ze stavu s_i v čase t do stavu s_j v čase $t+1$

$$a_{ij} = P(s(t+1) = s_j | s(t) = s_i). \tag{2.1}$$

Pravděpodobnost přechodu je v čase t generování akustické informace pro všechny stavy s_i



Obrázek 2.1: Příklad třístavového skrytého Markovova modelu používaného pro modelování trifónů, převzatý z [4].

konstantní a pro $i = 1, 2, \ldots, N-1$ platí:

$$\sum_{j=2}^{N} a_{ij} = 1, \tag{2.2}$$

kde N je celkový počet stavů modelu.

Výstupní pravděpodobnost $b_j(\boldsymbol{o}(t))$ popisuje rozdělení pravděpodobnosti pozorování $\boldsymbol{o}(t)$ produkovaného stavem s_j v čase t

$$b_j(\boldsymbol{o}(t)) = p(\boldsymbol{o}(t)|s(t) = s_j). \tag{2.3}$$

Ve stavech akustického modelu pro rozpoznávání plynulé řeči se v současné době nejvíce využívá normální rozdělení výstupní pravděpodobnosti reprezentované **modelem Gaussovských směsí** (GMM – Gaussian Mixture Model)

$$b_j(\boldsymbol{o}(t)) = \sum_{m=1}^{M} \omega_{jm} b_{jm}(\boldsymbol{o}(t)), \tag{2.4}$$

$$\text{kde} \qquad b_{jm}(\boldsymbol{o}(t)) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{C}_{jm}|}} \exp\left(-\frac{1}{2}(\boldsymbol{o}(t) - \boldsymbol{\mu}_{jm})^T \boldsymbol{C}_{jm}^{-1}(\boldsymbol{o}(t) - \boldsymbol{\mu}_{jm})\right), \tag{2.5}$$

$$\text{platí také} \qquad \int_{\boldsymbol{o}} b_j(\boldsymbol{o}) d\boldsymbol{o} = 1, \tag{2.6}$$

M značí počet složek hustotní směsi, n je dimenze kovarianční matice, ω_{jm}, $\boldsymbol{\mu}_{jm}$ a $\boldsymbol{C}_{jm}$ vyjadřují váhu, střední hodnotu a kovarianční matici normálního pravděpodobnostního rozložení m-té složky j-tého stavu modelu.

2.2 Výpočet pravděpodobnosti promluvy

Určení podmíněné pravděpodobnosti $P(\boldsymbol{O}|W)$ lze nahradit výpočtem $P(\boldsymbol{O}|\lambda)$ kde λ je skrytým Markovým modelem promluvy W. Výpočet pravděpodobnosti generování pozorované posloupnosti $\boldsymbol{O} = \{\boldsymbol{o}(1), \boldsymbol{o}(2), \ldots, \boldsymbol{o}(T)\}$ modelem λ, u něhož není známa posloupnost stavů $S = s(0), s(1)...s(T+1)$, kterými posloupnost pozorování prošla, lze počítat jako součet pravděpodobností všech možných posloupností stavů:

$$P(\boldsymbol{O}|\lambda) = \sum_S P(\boldsymbol{O}, S|\lambda) = \sum_S \left(a_{s(0)s(1)} \left[\prod_{t=1}^{T} b_{s(t)}(\boldsymbol{o}(t)) a_{s(t)s(t+1)} \right] \right), \tag{2.7}$$

kde $s(0)$ je vstupní neemitující stav a $s(T+1)$ výstupní neemitující stav modelu λ. Neemitující stavy jsou takové, které negenerují žádná pozorování a nemají tedy žádné k nim příslušné rozdělení pravděpodobnosti. Skryté modely modelují jednotlivé řečové jednotky, neemitující stavy slouží k pospojování těchto jednotek v jakoukoliv řečovou posloupnost.

Přímý výpočet $P(\boldsymbol{O}|\lambda)$ je výpočetně náročný, proto byl navrhnut **forward-backward iterační algoritmus**, který snižuje složitost výpočtu průběžným ukládáním mezivýsledků, které jsou poté použity pro všechny posloupnosti stavů z S se stejnou počáteční sekvencí stavů. Alternativou k výpočtu $P(\boldsymbol{O}|\lambda)$ jako součtu přes všechny možné cesty délky T modelem λ je aproximovat tuto sumu pouze jednou nejpravděpodobnější posloupností stavů, se kterou projde posloupnost $\boldsymbol{O}$ modelem λ, tj.

$$P_S(\boldsymbol{O}|\lambda) = \max_S P(\boldsymbol{O}, S|\lambda) = \max_S \left(a_{s(0)s(1)} \prod_{t=1}^{T} b_{s(t)}(\boldsymbol{o}(t)) a_{s(t)s(t+1)} \right). \tag{2.8}$$

Pro nalezení optimální posloupnosti stavů a vypočtení pravděpodobnosti $P_S(\boldsymbol{O}|\lambda)$ se využívá tzv. **Vitterbiův algoritmus** [7] pracující na principu dynamického programování.

2.2.1 Rekurzivní výpočet forward-backward algoritmem

Při výpočtu odpředu (forward) definujeme sdruženou pravděpodobnost $\alpha_j(t)$ pozorování posloupnosti prvních t akustických vektorů $\{\boldsymbol{o}(1), \ldots, \boldsymbol{o}(t)\}$ končící v aktuálním stavu s_j v čase t za podmínky modelu λ

$$\alpha_j(t) = P(\boldsymbol{o}(1), \boldsymbol{o}(2), \ldots, \boldsymbol{o}(t), s(t) = s_j|\lambda). \tag{2.9}$$

Pro výpočet odzadu (backward) definujeme podmíněnou pravděpodobnost $\beta_j(t)$ pozorování posloupnosti posledních $T-t$ akustických vektorů $\{\boldsymbol{o}(t+1), \boldsymbol{o}(t+2), \ldots, \boldsymbol{o}(T)\}$ za podmínky, že model λ je v čase t ve stavu s_j

$$\beta_j(t) = P(\boldsymbol{o}(t+1), \boldsymbol{o}(t+2), \ldots, \boldsymbol{o}(T)|s(t) = s_j, \lambda). \tag{2.10}$$

Konkrétní algoritmy výpočtu pravděpodobnosti $P(\boldsymbol{O}|\lambda)$ lze nalézt např. v [4]. Hledaná pravděpodobnost $P(\boldsymbol{O}|\lambda)$ může být snadno vyčíslena kombinací proměnných $\alpha_j(t)$ a $\beta_j(t)$

$$P(\boldsymbol{O}|\lambda) = \sum_{i=2}^{N-1} \alpha_i(t)\beta_i(t). \tag{2.11}$$

2.2.2 Iterativní Viterbiho algoritmus

Při procházení modelu si algoritmus uchovává proměnnou $\varphi_j(t)$ určující pravděpodobnost maximálně pravděpodobné posloupnosti stavů $s(1), s(2), \ldots, s(t) = s_j$ pro částečnou posloupnost pozorování $\{\boldsymbol{o}(1), \boldsymbol{o}(2), \ldots, \boldsymbol{o}(t)\}$

$$\varphi_j(t) = \max_{s(1),\ldots,s(t-1)} P(\boldsymbol{o}(1), \ldots, \boldsymbol{o}(t), s(1), \ldots, s(t) = s_j|\lambda). \tag{2.12}$$

Algoritmus postupuje odpředu, ale pro určení maximálně pravděpodobné posloupnosti stavů je potřeba si při jeho výpočtu ještě pamatovat v každém časovém kroku t, z kterého stavu v předchozím kroku byla vybrána maximální hodnota. K tomuto účelu je v algoritmu zavedena proměnná $\Psi_j(t)$, která se využívá při zpětném trasování k nalezení maximálně pravděpodobné cesty modelem λ pro posloupnost $\{\boldsymbol{o}(1), \boldsymbol{o}(2), \ldots, \boldsymbol{o}(T)\}$. Kompletní algoritmus lze nalézt např. v [4].

2.3 Trénování parametrů akustického modelu

Stanovení topologie skrytého Markovova modelu je úlohou expertního návrhu, vycházejícího z vlastností spojité řeči. Naopak ke stanovení parametrů modelu dochází na základě statistických metod aplikovaných na trénovací data, která jsou předem zanotována [4]. Parametry skrytého Markovova modelu jsou pravděpodobnosti přechodů a_{ij} a výstupní pravděpodobnosti $b_j(.)$ vyjádřené pomocí hustotní směsi normálního rozdělení s vahami ω_{jm}, středními hodnotami $\boldsymbol{\mu}_{jm}$ a kovariančními maticemi $\boldsymbol{C}_{jm}$

$$\lambda = \{a_{ij}, \omega_{jm}, \boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm}\}, \quad \text{kde } 1 \leq i, j \leq N \text{ a } 1 \leq m \leq M. \tag{2.13}$$

2.3.1 Metoda maximální věrohodnosti (ML)

Jako metoda odhadu parametrů bývá pro svou efektivitu často využívána **metoda maximální věrohodnosti** (ML – Maximum Likelihood), která maximalizuje výpočet pravděpodobnosti modelu

$$\lambda^* = \arg\max_{\lambda} P(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E|\lambda) \tag{2.14}$$

pro soubor E známých trénovacích promluv $\{\boldsymbol{O}^e\}_{e=1}^E$, kde $\boldsymbol{O^e} = \{\boldsymbol{o}^e(1), \boldsymbol{o}^e(2), \ldots, \boldsymbol{o}^e(T_e)\}$. Využívá se **Fisherova funkce věrohodnosti**

$$F(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E|\lambda) = P(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E|\lambda) = \prod_{e=1}^{E} P(\boldsymbol{O}^e|\lambda), \tag{2.15}$$

která je maximalizována přes neznámé parametry modelu λ (v praxi se spíše pracuje s logaritmem věrohodnostní funkce)

$$\hat{\lambda} = \arg\max_{\lambda} \log \prod_{e=1}^{E} P(\boldsymbol{O}^e|\lambda) = \arg\max_{\lambda} \sum_{e=1}^{E} \log P(\boldsymbol{O}^e|\lambda). \tag{2.16}$$

Pro stanovení optimálních parametrů modelu λ, tedy nalezení globálního maxima věrohodnostní funkce, v podstatě neexistuje žádná explicitní metoda. Efektivně se však k výpočtu využívá iterativního **Baum-Welchova** (BW) algoritmu [8], který je speciálním případem **EM** (EM – Expectation-Maximization) algoritmu [9]. EM algoritmus nalezne parametry modelu, které zabezpečí pouze lokální maximum funkce $P(\boldsymbol{O}|\lambda)$, výsledek tedy závisí na počáteční volbě parametrů.

EM algoritmus

Nejprve zavedeme skrytou proměnnou y^e, která ponese informaci o indexech stavů $s^e(t)$ a indexech složek hustotní směsi $m^e(t)$, tedy y^e je časová posloupnost dvojic $[s^e(t), m^e(t)], t = 1, \ldots, T_e$. Pak lze odvodit reestimační vztahy pro EM algoritmus ze vztahu:

$$P(\boldsymbol{O}^e|\lambda) = \sum_{y_e} P(\boldsymbol{O}^e, y^e|\lambda) = \sum_{y_e} P(y^e|\lambda)P(\boldsymbol{O}^e|y^e, \lambda) = \sum_{y_e} P(\boldsymbol{O}^e|\lambda)P(y^e|\boldsymbol{O}^e, \lambda). \tag{2.17}$$

Pokud uvažujeme rozdíl logaritmů věrohodnostních funkcí dvou modelů λ a $\bar{\lambda}$, platí po úpravě [4]:

$$\sum_{e=1}^{E} \log \frac{P(\boldsymbol{O}^e|\bar{\lambda})}{P(\boldsymbol{O}^e|\lambda)} = \sum_{e=1}^{E} \sum_{y^e} P(y^e|\boldsymbol{O}^e, \lambda) \log \left[\frac{P(\boldsymbol{O}^e, y^e|\bar{\lambda})}{P(\boldsymbol{O}^e, y^e|\lambda)} \frac{P(y^e|\boldsymbol{O}^e, \lambda)}{P(y^e|\boldsymbol{O}^e, \bar{\lambda})} \right]. \tag{2.18}$$

Vhodnou úpravou a aplikací nerovnosti $z \leq z - 1 (z \geq 0)$ dostáváme základní nerovnost EM algoritmu

$$\sum_{e=1}^{E} \log \frac{P(\boldsymbol{O}^e|\bar{\lambda})}{P(\boldsymbol{O}^e|\lambda)} \geq \sum_{e=1}^{E} \sum_{y_e} P(y^e|\boldsymbol{O}^e, \lambda) \log \frac{P(\boldsymbol{O}^e, y^e|\bar{\lambda})}{P(\boldsymbol{O}^e, y^e|\lambda)} = Q(\lambda, \bar{\lambda}) - Q(\lambda, \lambda), \tag{2.19}$$

$$\text{kde} \quad Q(\lambda, \bar{\lambda}) = \sum_{e=1}^{E} \sum_{y_e} P(y^e|\boldsymbol{O}^e, \lambda) \log P(\boldsymbol{O}^e, y^e|\bar{\lambda}). \tag{2.20}$$

Tato nerovnost říká, že pokud vybereme model $\bar{\lambda}$ tak, abychom dosáhli přírůstku funkce $Q(\lambda, \bar{\lambda})$ oproti funkci $Q(\lambda, \lambda)$, pak vzroste i logaritmus věrohodnostní funkce $\sum_{e=1}^{E} \log P(\boldsymbol{O}^e|\bar{\lambda})$. Výpočet EM algoritmu probíhá iterativně ve dvou krocích, nejprve vypočteme očekávání (expectation) funkce $Q(\lambda, \bar{\lambda})$ a následně vybereme takový model $\bar{\lambda}$, který maximalizuje (maximalization) funkci $Q(\lambda, \bar{\lambda})$. Odvození algoritmu lze nalézt mimo jiné v [10].

Rozepsáním pravděpodobnostní funkce pro jednotlivé parametry hustotních směsí modelu $\bar{\lambda}$ a dosazením do vztahu (2.20) dostáváme vztah pro přírůstkovou funkci $Q(\lambda, \bar{\lambda})$ s vyjádřenými parametry hustotních směsí

$$Q(\lambda, \bar{\lambda}) = \sum_{e=1}^{E} \sum_{y^e} P(y^e|\boldsymbol{O}^e, \lambda) \log P(\boldsymbol{O}^e, y^e|\bar{\lambda}) =$$

$$= \sum_{e=1}^{E} \frac{1}{P(\boldsymbol{O}^e|\lambda)} \sum_{y^e} P(\boldsymbol{O}^e, y^e|\lambda) \log \left[\prod_{t=1}^{T_e} (\bar{a}_{s^e(t-1)s^e(t)} + \bar{c}_{s^e(t)m_t^e} + \bar{b}_{s^e(t)m_t^e}(\boldsymbol{o}^e(t))) + \bar{a}_{s^e(T_e)s^e(T_e+1)} \right]. \tag{2.21}$$

Tuto rovnici použijeme k odvození vztahů pro trénování parametrů modelu.

Reestimační Baum-Welchův algoritmus

Jde o speciální případ EM algoritmu, platí pro něj tedy stejné vztahy, které byly odvozeny v předchozí sekci. Nově odhadnutý model $\bar{\lambda}$ v každém kroku (pomocí maximalizace funkce $Q(\lambda, \bar{\lambda})$) zvyšuje pravděpodobnost modelu $P(\boldsymbol{O}^e|\bar{\lambda}) \geq P(\boldsymbol{O}^e|\lambda)$ až do posledního kroku, kdy $P(\boldsymbol{O}^e|\bar{\lambda}) = P(\boldsymbol{O}^e|\lambda)$. Popis algoritmu lze nalézt například v [4].

2.3.2 Metoda maximální aposteriorní pravděpodobnosti (MAP)

Metoda maximální aposteriorní pravděpodobnosti (MAP – Maximum A-Posteriori Probability) [4] staví také na ML kritériu (viz část 2.3.1), rozdíl však je v uvažování λ jako náhodného vektoru a ne jako pevné hodnoty (jak je tomu v metodě ML). MAP kombinuje informaci získanou apriorním modelem λ s informací z trénovacích dat. Výhodou metody MAP je potřeba menšího množství trénovacích dat oproti metodě ML.

Úlohu nalezení parametrů λ lze formulovat na základě maximální pravděpodobnosti následovně:

$$\lambda^* = \arg\max_{\lambda} P(\lambda|\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E). \tag{2.22}$$

Využitím Bayesova pravidla dostáváme vztah:

$$\lambda^* = \arg\max_{\lambda} \frac{P(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E|\lambda)P(\lambda)}{P(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E)}. \tag{2.23}$$

Jmenovatel $P(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E)$ je pro všechny hodnoty λ konstantní, tady vztah (2.23) lze zjednodušit na tvar

$$\lambda^* = \arg\max_{\lambda} P(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E|\lambda)P(\lambda), \tag{2.24}$$

kde $P(\lambda)$ je apriorní informace rozdělení vektoru parametrů, což je jediná odlišnost od metody maximální věrohodnosti (2.14). Opět se využije Fisherova funkce věrohodnosti (2.15) jako při odvozování metodou ML.

Pro parametry diskrétních rozdělení, jako je případ pravděpodobností přechodu a_{ij} a vah hustotní směsi ω_{ij}, se jako apriorní hustota volí Dirichletovo rozdělení. Pro mnoharozměrné normální rozdělení s vektorem středních hodnot $\boldsymbol{\mu}$ a plnou kovarianční maticí $\boldsymbol{C}$ se volí apriorní hustota ve tvaru normálního-Wishartova rozdělení. Odvozené vztahy pro nové parametry modelu $\bar{\lambda}$ metodou MAP mají následující tvar:

$$\bar{a}_{1j} = \frac{(\eta_{1j} - 1) + \sum_{e=1}^{E} \frac{1}{P(\boldsymbol{O}^e|\lambda)} \alpha_j^e(1)\beta_j^e(1)}{\sum_{i=2}^{N-1}(\eta_{1i} - 1) + E}, \tag{2.25}$$

$$\bar{a}_{ij} = \frac{(\eta_{ij} - 1) + \sum_{e=1}^{E} \frac{1}{P(\boldsymbol{O}^e|\lambda)} \sum_{t=1}^{T_e-1} \alpha_i^e(t)a_{ij}b_j(\boldsymbol{o}^e(t))\beta_j^e(t+1)}{\sum_{i=2}^{N-1}(\eta_{1i} - 1) + \sum_{e=1}^{E} \frac{1}{P(\boldsymbol{O}^e|\lambda)} \sum_{t=1}^{T_e} \alpha_i^e(t)\beta_i^e(t)}, \tag{2.26}$$

$$\bar{a}_{iN} = \frac{(\eta_{1N} - 1) + \sum_{e=1}^{E} \frac{1}{P(\boldsymbol{O}^e|\lambda)} \alpha_i^e(T_e)\beta_i^e(T_e)}{\sum_{i=2}^{N-1}(\eta_{1i} - 1) + \sum_{e=1}^{E} \frac{1}{P(\boldsymbol{O}^e|\lambda)} \sum_{t=1}^{T_e} \alpha_i^e(t)\beta_i^e(t)}, \tag{2.27}$$

$$\bar{\omega}_{jm} = \frac{(\upsilon_{jm} - 1) + \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}^e(t)}{\sum_{\acute{m}=1}^{M} \left[(\upsilon_{j\acute{m}} - 1) + \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{j\acute{m}}^e(t)\right]}, \tag{2.28}$$

$$\bar{\boldsymbol{\mu}}_{jm} = \frac{\tau_{jm}\boldsymbol{\zeta}_{jm} + \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}^e(t)\boldsymbol{o}^e(t)}{\tau_{jm} + \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}^e(t)}, \tag{2.29}$$

$$\bar{\boldsymbol{C}}_{jm} = \frac{\boldsymbol{u}_{jm} + \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}^e(t)(\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})(\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})^T - \tau_{jm}(\bar{\boldsymbol{\mu}}_{jm} - \boldsymbol{\zeta}_{jm})(\bar{\boldsymbol{\mu}}_{jm} - \boldsymbol{\zeta}_{jm})^T}{\alpha_{jm} - n + \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}^e(t)}, \tag{2.30}$$

kde

$$\gamma_{jm}^{e}(t) = \gamma_{j}^{e}(t)\frac{\mathcal{N}(\boldsymbol{o}^{e}(t)|\boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm})}{\sum_{\acute{m}=1}^{M} c_{j\acute{m}}\mathcal{N}(\boldsymbol{o}^{e}(t)|\boldsymbol{\mu}_{j\acute{m}}, \boldsymbol{C}_{j\acute{m}})}. \tag{2.31}$$

Matice $\boldsymbol{u}_{jm}$ řádu n, vektor $\boldsymbol{\zeta}_{jm}$ a skaláry τ_{jm}, α_{jm} jsou parametry normálního-Wishartova aprioriního rozdělení m-té komponenty j-tého stavu a η_{ij}, υ_{jm} jsou složky vektorů parametrů Dirichletových aprioriních hustot pravděpodobností přechodů z i-tého do j-tého stavu HMM a vah m-té komponenty hustotní směsi j-tého stavu HMM[1]. Souhrnně se tyto parametry nazývají "hyperparametry"a reprezentují parametry aprioriního modelu. Nalezení hyperparametrů je složitý problém, jednou z možností je odhadovaní přímo z trénovacích dat [4].

2.3.3 Diskriminativní trénování (DT)

Nejpoužívanější přístup k trénování, ML kritérium, je vhodný pro rychlé vytvoření dobrého modelu využitím Baum-Welchova algoritmu. Tento generativní přístup vykazuje nejlepší vlastnosti za určitých předpokladů, které je však často velmi obtížné splnit. Jedním z nich je stacionarita řečového ústrojí v mikrosegmentech řeči, tedy že řeč je generována diskrétně. Druhou nesplnitelnou podmínkou je předpoklad nekonečného množství dat pro trénování [11].

Pro překonání těchto problémů byla navržena alternativní kritéria pro **diskriminativní trénování** (DT – Discriminative Training) HMM modelu. V mnoha odborných pracích bylo dokázáno (např. v [12]), že diskriminativní trénování může zlepšit úspěšnost rozpoznávání vytvořeného modelu formulováním funkce, která penalizuje parametry snižující správnost rozpoznávání. Diskriminativní trénování se snaží nastavit parametry modelu tak, aby jednotlivé stavy odpovídaly svým pozorováním s největší pravděpodobností a zároveň (na rozdíl od generativního trénování) minimalizuje pravděpodobnost pozorování patřících jiným stavům modelu.

Při optimalizování ML metody je vhodné použít EM algoritmus s pomocnou funkcí $Q(\lambda, \bar{\lambda})$ (2.20), kde zvýšení hodnoty této funkce garantuje nesnížení pravděpodobnosti $P(\boldsymbol{O}^{e}|\bar{\lambda})$. Funkce s touto vlastností je označována za *strong-sense* pomocnou funkci. Takovouto funkci je však pro kritéria DT obtížné najít. Optimalizace diskriminativního trénování se provádí **rozšířeným Baum-Welchovým algoritmem** (EBW – extended Baum-Welch) [13], který přidává do původní nerovnosti v BW algoritmu brzdící faktor, čímž zajistí konvexnost pomocné funkce, a optimalizaci diskriminativního trénování lze pak provést stejným způsobem jako v případě metody maximální věrohodnosti. Alternativním přístupem je využití *weak-sense* pomocné funkce, která však nezaručuje stabilitu odhadu kritéria DT. Je nutno zavést vhodný brzdící faktor. Více o brzdícím faktoru v následujících kapitolách zabývajících se diskriminativní adaptací 3.3.1,3.4.3.

Jednotlivá diskriminativní kritéria:

- **Maximalizace vzájemné informace** (MMI – Maximum Mutual Information)[14] umožňuje vybrat sekvenci slov s minimální nejistotou správné hypotézy. Tento přístup využívá informaci o správném přepisu promluvy $\boldsymbol{O}$ (tzv. referenční přepis $\boldsymbol{W_{ref}}$) a informaci o všech možných přepisech $\boldsymbol{W}$ (včetně toho správného). Toto kritérium lze napsat ve formě

$$\mathcal{F}_{MMI}(\lambda) = \frac{p^{\kappa}(\boldsymbol{O}|W_{ref}, \lambda)P(W_{ref})}{\sum_{W} p^{\kappa}(\boldsymbol{O}|W, \lambda)P(W)}, \tag{2.32}$$

  kde W_{ref} je referenční přepis nahrávky $\boldsymbol{O}$, zatímco W značí všechny možné přepisy, včetně toho správného. λ je HMM model. κ je empiricky volený faktor, kterým lze měnit

[1] Pro hodnoty $\eta_{ij} = 1, \upsilon_{jm} = 1, \boldsymbol{u}_{jm} = 0$ a $\alpha_{jm} = n$ nabývají vztahy (2.25) až (2.30) pro metodu MAP stejného tvaru jako rovnice pro metodu ML, tedy apriorní rozložení nenese žádnou informaci a odhad nových parametrů je proveden jen na základě trénovacích dat.

poměr mezi pravděpodobností správného přepisu a pravděpodobností ostatních přepisů, tedy lze jím regulovat míru diskriminativnosti výsledného modelu. V praxi jsou uvažovány místo všech možných přepisů $\boldsymbol{W}$ pouze N-nejlepší přepisy získané z rozpoznávače nebo N-nejpravděpodobnějších cest ze slovní mřížky. Podobné kritérium **Maximalizace vzájemné informace pomocí diskriminace pozorování** (MMI-FD – Maximum Mutual Information Frame Discrimination) [15] pracuje přímo s vektory pozorování a jejich příslušností ke stavům modelu namísto informací ze slovní mřížky.

- **Minimalizace chyby klasifikace** (MCE – Minimum Classification Error)[16] minimalizuje chybu očekávání přidáním ztrátové funkce $l(W, W_{ref})$ k diskriminativnímu kritériu

$$\mathcal{F}_{MCE}(\lambda) = \frac{p^{\kappa}(\boldsymbol{O}|W_{ref}, \lambda)P(W_{ref})}{\sum_W p^{\kappa}(\boldsymbol{O}|W, \lambda)P(W)} l(W, W_{ref}), \tag{2.33}$$

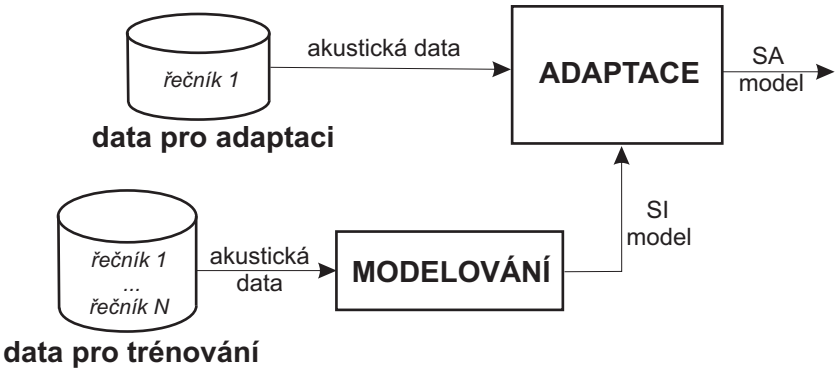
  kde opět W_{ref} je referenční přepis nahrávky $\boldsymbol{O}$, W značí všechny možné přepisy a κ je empiricky volený faktor. Možností jak vypočítat $l(W, W_{ref})$ je uvažovat **minimalizaci chyby fonému** (MPE – Minimum Phone Error) [12] nebo **minimalizaci chyby slova** (MWE – Minimum Word Error) [17].

Výše vyjmenované přístupy jsou vzájemně kombinovatelné, což přináší další zlepšení účinnosti akustického modelu [18]. Nevýhodou diskriminativního přístupu je potřeba většího množství dat pro trénování, než je potřeba pro klasické ML kritérium.

Kapitola 3

Metody adaptace

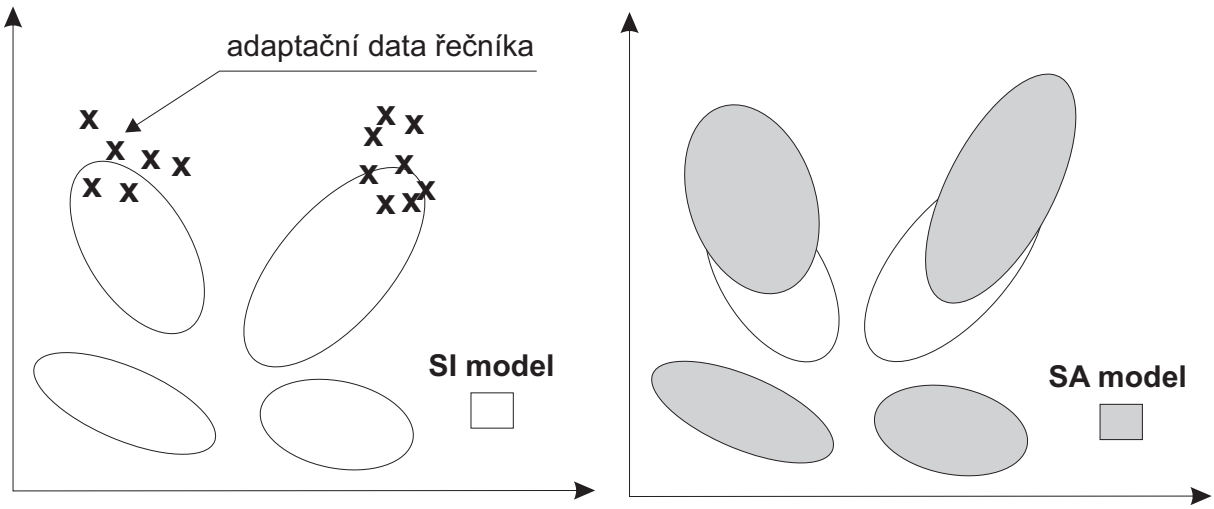
Skrytý Markovovův model (HMM – Hidden Markov Model) v kombinaci s **modelem Gaussovských směsí** (GMM – Gaussian Mixture Model) je již delší dobu nejlepším nástrojem pro účinné modelování akustických příznaků v úloze rozpoznávaní řeči [4]. Pro natrénování takového modelu je potřeba zpravidla velkého množství dat, což je obvykle nemožné získat od jednoho řečníka, proto se pro trénování modelu využívá dat od velkého množství řečníků. Výsledný akustický model, **na řečníku nezávislý** (SI – Speaker Independent), pak dovede rozpoznávat řeč obecného řečníka, protože trénovací data jsou v jistém smyslu průměrná.



Obrázek 3.1: Schématické znázornění adaptace.

Pokud je však totožnost řečníka při rozpoznávání známá, bylo by možné dosáhnout větší úspěšnosti natrénováním modelu jenom z dat konkrétního řečníka, kterého budeme chtít rozpoznávat. Takovému modelu se pak říká **na řečníku závislý** (SD – Speaker Dependent). Problémem při tvorbě SD modelu je nutnost mít k dispozici velký počet trénovacích promluv od jednoho řečníka. Řešení poskytuje adaptace SI modelu na data konkrétního řečníka, vzniklý model je **na řečníka adaptovaný** (SA – Speaker Adaptive), viz obr. 3.1. Jde vlastně o transformaci SI modelu ve smyslu dosažení maximální pravděpodobnosti pro nová data, viz obrázek 3.2.

Na rozdíl od vlastního trénování akustického modelu využívá adaptace apriorní znalost o rozložení parametrů akustického modelu. Tato znalost je obvykle odvozována z předem natrénovaného SI modelu. Adaptace přizpůsobuje SI model tak, aby byla maximalizována pravdě-



Obrázek 3.2: Ilustrativní příklad adaptace modelu. Hustoty rozložení složek SI modelu (zde reprezentovány elipsou) se "posunou"ve směru adaptačních dat tak, aby SA model tato data lépe modeloval.

podobnost adaptačních dat:

$$\lambda^* = \arg\max_{\lambda} P(\boldsymbol{O}^1, \ldots, \boldsymbol{O}^E|\lambda)P(\lambda), \tag{3.1}$$

kde $P(\lambda)$ představuje apriorní informaci o rozdělení vektoru parametrů modelu λ (dána obvykle SI modelem), $\boldsymbol{O}^e = \{\boldsymbol{o}^e(1), \boldsymbol{o}^e(2), \ldots, \boldsymbol{o}^e(T_e)\}, e = 1, \ldots, E$, je posloupnost vektorů příznaků přidružených jedinému řečníkovi a λ^* je nejlepším odhadem parametrů SA modelu tohoto řečníka.

3.1 Obecné dělení adaptačních metod

Adaptačních přístupů a z nich vyplývajících různých metod k adaptaci je velké množství. Obecně je možné dělit tyto metody z hlediska několika kritérií podle jejich vlastností:

- Adaptace může probíhat buď **za chodu aplikace** (*on-line*) - podrobněji popsáno v kapitole 5, nebo může být provedena **před vlastním testováním** (*off-line*) - na tento případ nejsou kladeny žádné speciální požadavky, konkrétně rychlost adaptace zde nehraje velkou roli.

- Pokud máme při adaptaci k dispozici přesný fonetický přepis adaptačních dat, značíme úlohu za adaptaci **s učitelem** (*supervised*). Pokud však přesný přepis nemáme, tzv. adaptace **bez učitele** (*unsupervised*), lze jej nahradit automatickým přepisem pomocí SI modelu. Výsledný přepis obvykle obsahuje nepřesnosti a chyby, které lze odstranit například využitím adaptovaného modelu v další iteraci (zpřesňujeme přepis a tím i SA model), popřípadě uvažováním **faktoru jistoty** (CF – Certainty Factor) [19] přepsaných slov jako výstupu z jazykového modelu (bereme jen slova, která se rozpoznala s dostatečně velkou jistotou). Problémy unsupervised adaptace je nutné řešit převážně při on-line aplikacích, proto je tato úloha podrobněji popsána v podkapitole 5.1.

- Adaptační metody lze dělit podle toho, zda **transformují parametry modelu** (*model transformation*) nebo **transformují vektory pozorování** (*feature transformation*). Druhá možnost má výhodu v malých paměťových nárocích, protože si není třeba pamatovat pro každého řečníka celý model, ale jen transformaci, která transformuje konkrétní data pro lepší rozpoznání SI modelem.

- Pokud jsou při adaptaci použita všechna data najednou, jedná se o **dávkovou** (*batch*) adaptaci. Pokud se však systém adaptuje postupně, jak přicházejí nová adaptační data, jde o **inkrementální** (*incremental*) adaptaci, která se nejčastěji používá v on-line systémech (podrobněji o ní v podkapitole 5.2).

- Pro vygenerování SA modelu lze použít přístup **generativní** (*generative*) adaptace, kdy složky modelu nejlépe reprezentují příslušná data. Jiným přístupem je **diskriminativní** (*discriminative* adaptace, kdy složky SA modelu nejlépe reprezentují svá data, ale navíc se co nejméně vzájemně překrývají.

- Při určování efektivity adaptačních metod je nutné uvažovat také **množství dat**, které jsou pro adaptaci k dispozici. Pro ideální adaptační metodu platí: Adaptovaný SA model konverguje k modelu SD konkrétního řečníka při dostatečném množství adaptačních dat (množství, které by bylo potřebné pro vlastní natrénování SD modelu) a zároveň pro menší počet adaptačních dat je adaptace rychlá a přitom je dobrou aproximací SD modelu. Obvykle jsou tyto dva předpoklady ve vzájemném rozporu.

Experimenty na klasických metodách adaptace popisovaných dále v této kapitole lze nalézt v podkapitole 7.4 společně se srovnáním výsledků jednotlivých přístupů k adaptaci podle výše uvedeného rozdělení.

3.2 Akumulované statistiky

Parametry, které nesou nejdůležitější informaci o řečníkovi, jsou střední hodnoty a kovarianční matice výstupních pravděpodobností stavů HMM tvořených GMM. Adaptační metody potřebují ke své správné funkčnosti dostatečně velký vzorek dat od adaptovaného řečníka, avšak přistupovat k datům v průběhu adaptace by bylo časově náročné, proto většina adaptačních technik pracuje pouze s naakumulovanými statistikami adaptačních dat uvedenými níže. Následující vzorce jsou pro jednotlivé adaptační metody společné a v dalším textu na ně bude odkazováno.

Nechť

$$\gamma_{jm}^{e}(t) = \frac{\omega_{jm} p(\boldsymbol{o}^{e}(t)|jm)}{\sum_{m=1}^{M} \omega_{jm} p(\boldsymbol{o}^{e}(t)|jm)} \tag{3.2}$$

je aposteriorní pravděpodobnost, že pozorování $\boldsymbol{o}(t)$ je generováno m-tou složkou Gaussovské směsi j-tého stavu HMM. ω_{jm}, $\boldsymbol{\mu}_{jm}$ a $\boldsymbol{C}_{jm}$ je váha, střední hodnota a kovarianční matice m-té složky v j-tém stavu HMM. Dále lze definovat

$$c_{jm} = \sum_{e=1}^{E} \sum_{t=1}^{Te} \gamma_{jm}^{e}(t) \tag{3.3}$$

obsazení m-té složky v j-tém stavu HMM přes všechny časy t a vektor

$$\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}) = \frac{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}^{e}(t) \boldsymbol{o}^{e}(t)}{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}^{e}(t)} \tag{3.4}$$

resp.

$$\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\cdot\boldsymbol{o}^{\boldsymbol{T}}) = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^{e}(t)\boldsymbol{o}^{e}(t)\boldsymbol{o}^{eT}(t)}{\sum_{e=1}^{E}\sum_{t=1}^{Te}\gamma_{jm}^{e}(t)} \tag{3.5}$$

jako první a druhý statistický moment hodnot příznaků přiřazených k m-té složce GMM v j-tém stavu HMM. Přiřazení je dáno tzv. zarovnáním dat do jednotlivých stavů HMM modelu (*force-alignment*) a poté rozdistribuováním mezi složky daného stavu s uvažováním jejich váhy.

Při uvažování některých kritérií používaných v diskriminativním trénování v podkapitole 2.3.3, jako je například MMI (2.32), je nutno nasčítávat ještě doplňkové, tzv. *den* statistiky (jsou počítány pomocí jmenovatele kritéria (2.32)):

$$\gamma_{jm}^{den}(t) = \sum_{j=1}^{J}\sum_{m=1}^{M}\frac{\omega_{jm}p(\boldsymbol{o}^{e}(t)|jm)}{\sum_{m=1}^{M}\omega_{jm}p(\boldsymbol{o}^{e}(t)|jm)}, \tag{3.6}$$

$$c_{jm}^{den} = \sum_{e=1}^{E}\sum_{t=1}^{Te}\gamma_{jm}^{den}(t), \tag{3.7}$$

$$\boldsymbol{\varepsilon}_{jm}^{den}(\boldsymbol{o}) = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^{den}(t)\boldsymbol{o}^{e}(t)}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^{den}(t)}, \tag{3.8}$$

resp.

$$\boldsymbol{\varepsilon}_{jm}^{den}(\boldsymbol{o}\cdot\boldsymbol{o}^{\boldsymbol{T}}) = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^{den}(t)\boldsymbol{o}^{e}(t)\boldsymbol{o}^{eT}(t)}{\sum_{e=1}^{E}\sum_{t=1}^{Te}\gamma_{jm}^{den}(t)}, \tag{3.9}$$

3.3 Metoda maximální aposteriorní pravděpodobnosti (MAP)

Metoda **maximální aposteriorní pravděpodobnosti** (MAP – Maximum A-Posteriori Probability) je založena na Bayesově metodě odhadu parametrů akustického modelu s jednotkovou ztrátovou funkcí [20]. Nástin odvození metody byl uveden v podkapitole 2.3.2, kde byly také uvedeny vztahy pro přepočet nových parametrů modelu $\bar{\lambda}$, který maximalizuje funkci $Q(\bar{\lambda},\lambda)$. V případě adaptace odpadá problém hledání apriorních parametrů (tzv. hyperparametrů). Jako apriorní model je brán v úvahu právě námi adaptovaný SI model. Zbylé hyperparametry mají význam experimentálně určené adaptační konstanty τ. V praxi je výhodné adaptovat především vektory středních hodnot $\boldsymbol{\mu}_{jm}$, popřípadě i kovarianční matice $\boldsymbol{C}_{jm}$ a váhy ω_{jm} jednotlivých složek hustotních směsí modelu, zbylé parametry zůstávají totožné s apriorním modelem.

Z (2.25) až (2.27) lze odvodit následující vztahy pro MAP adaptaci:

$$\bar{\omega}_{jm} = \left[\frac{\alpha_{jm}c_{jm}}{T} + (1-\alpha_{jm})\omega_{jm}\right]\chi \quad , \tag{3.10}$$

$$\bar{\boldsymbol{\mu}}_{jm} = \alpha_{jm}\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}) + (1-\alpha_{jm})\boldsymbol{\mu}_{jm} \quad , \tag{3.11}$$

$$\bar{\boldsymbol{C}}_{jm} = \alpha_{jm}\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\cdot\boldsymbol{o}^{T}) + (1-\alpha_{jm})(\boldsymbol{C}_{jm} + \boldsymbol{\mu}_{jm}\boldsymbol{\mu}_{jm}^{T}) - \bar{\boldsymbol{\mu}}_{jm}\bar{\boldsymbol{\mu}}_{jm}^{T} \quad , \tag{3.12}$$

$$\alpha_{jm} = \frac{c_{jm}}{c_{jm}+\tau} \quad , \tag{3.13}$$

kde c_{jm} a $\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o})$ jsou definovány vztahy (3.3), respektive (3.4). χ je normalizační parametr, který garantuje, že všechny adaptované váhy každého GMM budou v součtu rovny jedné. α_{jm} je adaptační koeficient, který kontroluje rovnováhu mezi starými a novými parametry. K tomu je využívána empiricky určená konstanta τ, která nám říká, jak moc se mají staré

parametry posunout ve směru nových parametrů určených z adaptačních dat. Čím více dat k danému parametru máme, tím méně se původní hodnota projeví na výsledku. Adaptovaný model metodou MAP konverguje k výsledku získanému klasickým trénováním pro dostatečné množství dat. Nevýhodou MAP adaptace je, že informaci z adaptačních dat nijak nezobecňuje, tedy při malém počtu adaptačních dat pro konkrétní parametr modelu se adaptace pro tento parametr nijak neprojeví.

3.3.1 Diskriminativní MAP (DMAP)

Klasická metoda MAP je založena na kritériu ML (2.14). Takto adaptovaný model trpí stejnými problémy, které byly zmíněny v úvodu do diskriminativního trénování v podkapitole 2.3.3. Metoda **diskriminativní MAP** (DMAP – Discriminative MAP) naproti tomu staví na některých z kritérií definovaných pro diskriminativní trénování, jako je například v [21] kritérium MMI (2.32). Maximalizováním MMI kritéria zabezpečíme rostoucí pravděpodobnost pro správné přepisy, zatímco pravděpodobnost pro ostatní přepisy se bude snižovat, což vede k diskriminativnímu charakteru adaptace.

Pomocí MMI kritéria lze odvodit vztahy pro DMAP adaptaci. Na rozdíl od klasického MAP se nasčítávají ke statistikám (3.2), (3.3), (3.4) a (3.5) i tzv. *den* statistiky (3.6), (3.7), (3.8) a (3.9) odvozené z čitatele MMI kritéria, tedy počítány s využitím všech možných přepisů.

Pro DMAP je pak nutno pravděpodobnostní momenty (3.3), (3.4) a (3.5) nahradit rozdílem původní hodnoty a *den* hodnoty, $\gamma_{jm}(t)$ je nahrazeno $\gamma_{jm}^{den}(t)$. Poté je např. nová střední hodnota dána vztahem

$$\bar{\boldsymbol{\mu}}_{jm} = \frac{\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}) - f\boldsymbol{\varepsilon}_{jm}^{den}(\boldsymbol{o}) + \tau_{jm}\boldsymbol{\mu}_{jm}}{c_{jm} - fc_{jm}^{den} + \tau_{jm}}, \tag{3.14}$$

kde f reprezentuje brzdící konstantu pro udržení stability odhadu MMI kritéria.

Oproti klasické metodě MAP očekává diskriminativní přístup kvůli akumulaci *den* statistik větší počet adaptačních dat. Protože vlastní diskriminativní odhad je pomocí brzdícího faktoru tlumen směrem k ML odhadu, je vhodné adaptaci provést v několika následných iteracích. Podrobnější odvození DMAP pro diskriminativní kritérium MMI i MPE lze nalézt např. v [21],[22].

3.4 Metody adaptace založené na lineární transformaci (LT)

Základním nedostatkem adaptační metody MAP je potřeba dostatečného množství dat pro každý parametr akustického modelu. Jelikož je adaptovaných parametrů v modelu velmi mnoho, metoda vyžaduje nemalé množství adaptačních nahrávek, kterých se nám často nedostává. Metody založené na **lineárních transformacích** (LT – Linear Transformation) [23] omezují počet volných parametrů modelu shlukováním akusticky podobných složek stavů do tříd C_n, které pak adaptují stejným způsobem. Díky shlukování složek poskytují tyto metody dobré výsledky i s relativně malým počtem adaptačních dat (v porovnání s MAP) a samotná adaptace pak může být mnohem rychlejší. Metody se snaží pro každý shluk nalézt takovou lineární transformaci, kdy by adaptované parametry akustického modelu lépe odpovídaly hlasu konkrétního řečníka. Všechny parametry v jednom shluku se pak adaptují stejnou lineární transformací. Pro výpočet transformace je pak dostatek dat a adaptačními daty nepokryté parametry modelu jsou také zadaptovány. Více o shlukování parametrů v podkapitole 3.4.4.

V této práci rozlišujeme dva způsoby lineárních transformací modelu, a to **neomezenou** (*unconstrained*) a **omezenou** (*constrained*) transformaci. První z nich používá jiné transfor-

mační vztahy pro střední hodnoty a jiné pro kovarianční matice, na rozdíl od druhého způsobu, kde jsou tyto parametry transformovány stejnou transformační maticí. Dále lze u každé metody rozlišit, zda je adaptace zaměřena na **transformaci parametrů modelu** nebo na **transformaci příznaků pozorování**.

3.4.1 Metoda maximální věrohodné lineární regrese (MLLR)

Nejčastěji používaná adaptační technika ze skupiny lineárních transformací je metoda **maximální věrohodné lineární regrese** (MLLR – Maximum Likelihood Linear Regression) [24]. Metoda je založena na neomezené transformaci, tedy střední hodnoty a kovarianční matice jsou transformovány různými transformacemi. Předpokládejme opět adaptační data ve formě $\boldsymbol{O}^e = \{\boldsymbol{o}^e(1), \boldsymbol{o}^e(2), \ldots, \boldsymbol{o}^e(T_e)\}, e = 1, \ldots, E$.

Lineární transformace střední hodnoty je dána:

$$\bar{\boldsymbol{\mu}}_{jm} = \boldsymbol{A}_{(n)}\boldsymbol{\mu}_{jm} + \boldsymbol{b}_{(n)} = \boldsymbol{W}_{(n)}\boldsymbol{\xi}_{jm}, \tag{3.15}$$

kde $\boldsymbol{\mu}_{jm}$ je původní střední hodnota m-té složky GMM v j-tém stavu Si modelu, $\bar{\boldsymbol{\mu}}_{jm}$ je adaptovaná střední hodnota, $\boldsymbol{\xi}_{jm}^T = [\boldsymbol{\mu}_{jm}^T, 1]$ je původní střední hodnota rozšířená o 1, $\boldsymbol{A}_{(n)}$ je transformační matice a $\boldsymbol{b}_{(n)}$ je aditivní vektor, $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$ je transformační matice pro třídu C_n.

Transformace kovarianční matice je vyjádřena vztahem:

$$\bar{\boldsymbol{C}}_{jm} = \boldsymbol{L}\boldsymbol{H}_{(n)}\boldsymbol{L}^T, \tag{3.16}$$

kde $\boldsymbol{H}_{(n)}$ je transformační matice pro třídu C_n a $\boldsymbol{L}$ je Choleskiho faktor původní kovarianční matice $\boldsymbol{C}_{jm}$. Ekvivalentně lze vztah (3.16) zapsat ve tvaru

$$\bar{\boldsymbol{C}}_{jm} = \boldsymbol{H}_{(n)}\boldsymbol{C}_{jm}\boldsymbol{H}_{(n)}^T. \tag{3.17}$$

Úloha nalezení lineárních transformačních matic je vázána na nalezením optima následující funkce:

$$Q(\lambda, \bar{\lambda}) = const - \frac{1}{2}\sum_{b_{jm}\in\lambda}\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}(t)(c_{jm} + \log|\bar{\boldsymbol{C}}_{jm}| + (\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})^T\bar{\boldsymbol{C}}_{jm}^{-1}(\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})). \tag{3.18}$$

Implementačně lze rozdělit úlohu na dvě části:

- nalezení transformací pro střední hodnoty (3.15) (MLLRmean),
- nalezení transformací pro kovarianční matice (3.16) nebo (3.17) (MLLRcov).

Metoda MLLR pro střední hodnoty (MLLRmean)

Naším úkolem je nalézt matici $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$, která transformuje střední hodnoty všech gaussovských složek b_{jm} patřících do třídy C_n, tedy maximalizovat optimalizační funkci (3.18) [25]. Provedením derivace a vhodnou úpravou (3.18) lze dostat vztah

$$\sum_{b_{jm}\in C_n}\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^e(t)\boldsymbol{C}_{jm}^{-1}\boldsymbol{o}^e(t)\boldsymbol{\xi}_{jm}^T = \sum_{b_{jm}\in C_n}\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^e(t)\boldsymbol{C}_{jm}^{-1}\boldsymbol{W}_{(n)}\boldsymbol{\xi}_{jm}\boldsymbol{\xi}_{jm}^T. \tag{3.19}$$

Výraz (3.19) je možné pro lepší názornost přepsat zavedením substituční matice $\boldsymbol{Z}_{(n)}$ za celou levou část rovnice a $\boldsymbol{V}_{jm}\boldsymbol{D}_{jm}$ za pravou část uvnitř sumy přes složky shluku C_n. Zredukovaný tvar rovnice je pak

$$\boldsymbol{Z}_{(n)} = \sum_{b_{jm} \in C_n} \boldsymbol{V}_{jm}\boldsymbol{W}_{(n)}\boldsymbol{D}_{jm}. \tag{3.20}$$

Řešení rovnice (3.20) je výpočetně náročné, proto se v praxi více využívá výpočet přes řádky matice $\boldsymbol{W}_{(n)}$, který předpokládá akustický model s diagonálními kovariančními maticemi. Je-li matice $\boldsymbol{C}_{jm}$ diagonální (lze ji nahradit vektorem $\boldsymbol{\sigma}^2_{jm} = diag(\boldsymbol{C}_{jm})$), pak je diagonální i matice $\boldsymbol{V}_{jm}$. i-tý řádek matice $\boldsymbol{W}_{(n)}$ lze pak spočítat pro všechna $i = 1, \ldots, I$ ze vztahu

$$\boldsymbol{w}_i^T = \boldsymbol{z}_i^T \boldsymbol{G}_i^{-1} \tag{3.21}$$

kde

$$\boldsymbol{G}_i = \sum_{b_{jm} \in C_n} \frac{1}{\sigma_{jm}(i)^2} \boldsymbol{\xi}_{jm}\boldsymbol{\xi}_{jm}^T \sum_{e=1}^{E}\sum_{t=1}^{T_e} \gamma_{jm}^e(t). \tag{3.22}$$

Nepatrně odlišné odvození výpočtu transformační matice $\boldsymbol{W}_{(n)}$ za předpokladu diagonálních kovariančních matic lze nalézt v [24]. Uvedeno je zde pro lepší návaznost na odvození vztahů pro omezenou transformaci fMLLR (feature MLLR) popsanou v podkapitole 3.4.2. Odvození využívá vztahy (3.2), (3.3), (3.4) definované na začátku této kapitoly.

Část optimalizační funkce (3.18), která je závislá na $\boldsymbol{W}_{(n)}$, je:

$$Q_{\boldsymbol{W}_{(n)}} = const - \sum_{b_{jm} \in C_n} c_{jm} \sum_{i=1}^{I} \frac{(\boldsymbol{w}_{(n)i}^T \boldsymbol{\xi}_{jm})^2 - 2(\boldsymbol{w}_{(n)i}^T \boldsymbol{\xi}_{jm})\varepsilon_{jm}(o)(i)}{\sigma_{jm}^2(i)}. \tag{3.23}$$

Rovnice (3.23) může být dále přepsána na tvar:

$$Q_{\boldsymbol{W}_{(n)}} = \boldsymbol{w}_{(n)i}^T \boldsymbol{k}_{(n)i} - 0.5\boldsymbol{w}_{(n)i}^T \boldsymbol{G}_{(n)i}\boldsymbol{w}_{(n)i}, \tag{3.24}$$

kde

$$\boldsymbol{k}_{(n)i} = \sum_{b_{jm} \in C_n} \frac{c_{jm}\boldsymbol{\xi}_{jm}\boldsymbol{\varepsilon_{jm}(o)}(i)}{\sigma_{jm}^2(i)} \tag{3.25}$$

a

$$\boldsymbol{G}_{(n)i} = \sum_{b_{jm} \in C_n} \frac{c_{jm}\boldsymbol{\xi}_{jm}\boldsymbol{\xi}_{jm}^T}{\sigma_{jm}^2(i)}, \tag{3.26}$$

Pak maximalizováním rovnice (3.24) dostáváme:

$$\boldsymbol{w}_{(n)i} = \boldsymbol{G}_{(n)i}^{-1}\boldsymbol{k}_{(n)i}. \tag{3.27}$$

Metoda MLLR pro kovarianční matice (MLLRcov)

Tato metoda [25] se počítá ve dvou krocích, nejprve transformujeme střední hodnoty (stejný postup jako u metody MLLRmean), poté kovarianční matice. Postupně získáváme modely $\lambda = \{\boldsymbol{\mu}, \boldsymbol{C}\}, \bar{\lambda} = \{\bar{\boldsymbol{\mu}}, \boldsymbol{C}\}, \tilde{\lambda} = \{\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{C}}\}$ a platí pro ně: $p(\boldsymbol{O}|\lambda) \leq p(\boldsymbol{O}|\bar{\lambda}) \leq p(\boldsymbol{O}|\tilde{\lambda})$.

Jak již bylo zmíněno, lze transformaci kovarianční matice spočítat dvěma způsoby. První vychází z rovnice (3.16), kde $\boldsymbol{L}$ je získáno Choleskiho rozkladem matice $\boldsymbol{C} = \boldsymbol{L}\boldsymbol{L}^T$. Pak nejlepší odhad transformační matice $\boldsymbol{H}_{(n)}$ lze získat [23]

$$\boldsymbol{H}_{(n)} = \frac{\sum_{b_{jm}\in C_n}\left((\boldsymbol{L}_{jm}^{-1})^T\left[\sum_{e=1}^{E}\sum_{t=1}^{T_e} y_{jm}^{e}(t)(\boldsymbol{o}^e(t)-\bar{\boldsymbol{\mu}}_{jm})(\boldsymbol{o}^e(t)-\bar{\boldsymbol{\mu}}_{jm})\right]\boldsymbol{L}_{jm}^{-1}\right)}{\sum_{b_{jm}\in C_n}\sum_{e=1}^{E}\sum_{t=1}^{T_e} y_{jm}^{e}(t)}. \tag{3.28}$$

Rozpoznávání s takto adaptovaným modelem je značně výpočetně náročné (pokud uvažujeme plné kovarianční matice), protože logaritmus věrohodnosti $\mathcal{L}$ vektoru pozorování $\boldsymbol{o}(t)$ daný transformovaným modelem $\tilde{\lambda}$ je počítán jako logaritmus normálního rozdělení $\mathcal{N}$

$$\log\mathcal{L}(\boldsymbol{o}^e(t),\boldsymbol{\mu},\boldsymbol{C},\boldsymbol{W}_(n),\boldsymbol{H}_(n)) = \log\mathcal{N}(\boldsymbol{o}^e(t),\bar{\boldsymbol{\mu}},\bar{\boldsymbol{C}}), \tag{3.29}$$

kde $\bar{\boldsymbol{C}}$ bude nadále plná kovarianční matice, $\boldsymbol{W}_(n)$ a $\boldsymbol{H}_(n)$ je transformační funkce získané při adaptaci MLLRmean a MLLRcov.

Pokud však předpokládáme původní kovarianční matice modelu diagonální, je efektivnější vycházet ze vztahu (3.17) a počítat transformační matici po řádcích. Vektor $\boldsymbol{\sigma}_{jm}^2 = diag(\boldsymbol{C}_{jm})$ nahrazuje diagonální kovarianční matici $\boldsymbol{C}_{jm}$. i-tý řádek transformační matice $\boldsymbol{H}(n)$, tedy $\boldsymbol{h}_{(n)i}$, lze iterativně vypočítat jako

$$\boldsymbol{h}_{(n)i}^{-1} = \boldsymbol{v}_{(n)i}\boldsymbol{G}_{(n)i}^{-1}\sqrt{\frac{\sum_{jm\in C_n}\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^{e}(t)}{\boldsymbol{v}_{(n)i}\boldsymbol{G}_{(n)i}^{-1}\boldsymbol{C}_{jm}^{T}(i)}}, \tag{3.30}$$

kde

$$\boldsymbol{G}_{(n)i} = \sum_{jm\in C_n}\frac{1}{\sigma_{jm}^2(i)}\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{jm}^{e}(t)(\boldsymbol{o}^e(t)-\boldsymbol{\mu}_{jm})(\boldsymbol{o}^e(t)-\boldsymbol{\mu}_{jm})^T \tag{3.31}$$

a $\boldsymbol{v}_{(n)i}$ je kofaktor matice $\boldsymbol{H}_{(n)}^{-1}$.

Alternativní výpočet výsledného logaritmu věrohodnosti $\mathcal{L}$ pro konkrétní Gausian ze třídy C_n může být nyní počítán

$$\log\mathcal{L}(\boldsymbol{o}^e(t)|\boldsymbol{\mu}_{jm},\boldsymbol{C}_{jm},\boldsymbol{W}_{(n)},\boldsymbol{H}_{(n)}) = \log\mathcal{N}(\boldsymbol{H}_{(n)}^{-1}\boldsymbol{o}^e(t);\boldsymbol{H}_{(n)}^{-1}\bar{\boldsymbol{\mu}}_{jm},\boldsymbol{C}_{jm}) - 0.5\log(|\boldsymbol{H}_{(n)}|^2), \tag{3.32}$$

Vztah 3.32 je méně výpočetně náročný než vztah 3.29, protože není zapotřebí inverze matice $\boldsymbol{H}_(n)$ ani dvojitého násobení kovarianční matice $\boldsymbol{C}$ transformační maticí $\boldsymbol{H}_(n)$, viz (3.16) .

3.4.2 Metoda MLLR pro transformace vektorů pozorování (fMLLR)

Metoda **maximální věrohodné lineární regrese vektorů pozorování** (fMLLR – feature Maximum Likelihood Linear Regression) [23] je zaměřena na lineární transformaci vektoru příznaků $\boldsymbol{O}$, spíše než na transformaci samotného akustického modelu. To přináší výhody převážně v rychlosti adaptace (není potřeba transformovat rozsáhlý model s tisíci příznaky) a v paměťové náročnosti (pamatujeme si pouze transformaci, nikoliv celý nový model pro každého z řečníků). Transformace modelu metodou fMLLR je však v zásadě možná pouhým přepisem transformačních vztahů do jiné formy (viz níže), pak je metoda nazývána **omezenou MLLR** (CMLLR – Constrained Maximum Likelihood Linear Regression). Metoda fMLLR (nebo její ekvivalent pro transformaci modelu CMLLR) je omezenou transformací (viz obecné dělení LT metod v úvodu do podkapitoly 3.4), tedy střední hodnoty a kovarianční matice jsou transformovány stejnou transformací $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$

$$\bar{\boldsymbol{o}}^e(t) = \boldsymbol{A}_{(n)}\boldsymbol{o}^e(t) + \boldsymbol{b}_{(n)} = \boldsymbol{A}_{(n)c}^{-1}\boldsymbol{o}^e(t) + \boldsymbol{A}_{(n)c}^{-1}\boldsymbol{b}_{(n)c} = \boldsymbol{W}_{(n)}\boldsymbol{\xi}^e(t), \tag{3.33}$$

kde $\boldsymbol{\xi}^{eT}(t) = [\boldsymbol{o}^{eT}(t), 1]$ je rozšířený vektor příznaků a $\boldsymbol{A}_{(n)c}, \boldsymbol{b}_{(n)c}$ jsou matice pro ekvivalentní transformaci parametrů akustického modelu

$$\bar{\boldsymbol{\mu}}_{jm} = \boldsymbol{A}_{(n)c}\boldsymbol{\mu}_{jm} - \boldsymbol{b}_{(n)c}, \tag{3.34}$$

a

$$\bar{\boldsymbol{C}}_{jm} = \boldsymbol{A}_{(n)c}\boldsymbol{C}_{jm}\boldsymbol{A}_{(n)c}^{T}, \tag{3.35}$$

Optimalizační funkce pro odhad transformací nabývá tvaru:

$$Q(\lambda, \bar{\lambda}) = const - \frac{1}{2}\sum_{b_{jm}\in\lambda}\sum_{t,e=1}^{T_E}\gamma_{jm}^{e}(t)(c_{jm} + \log|\boldsymbol{C}_{jm}| - \log(|\boldsymbol{A}_{(n)}|^2) + (\bar{\boldsymbol{o}}^{e}(t) - \boldsymbol{\mu}_{jm})^T C_{jm}^{-1}(\bar{\boldsymbol{o}}^{e}(t) - \boldsymbol{\mu}_{jm})). \tag{3.36}$$

Analogicky jako v odvození pro metodu MLLRmean v podkapitole 3.4.1 lze optimalizační funkci (3.36) upravit na tvar [24]

$$Q_{\boldsymbol{W}_{(n)}}(\lambda, \bar{\lambda}) = \log(|\boldsymbol{A}_{(n)}|) + \sum_{i=1}^{I}\boldsymbol{w}_{(n)i}^{T}\boldsymbol{k}_i - 0.5\boldsymbol{w}_{(n)i}^{T}\boldsymbol{G}_{(n)i}\boldsymbol{w}_{(n)i}, \tag{3.37}$$

kde

$$\boldsymbol{k}_{(n)i} = \sum_{jm\in C_n}\frac{c_{jm}\boldsymbol{\mu}_{jm}(i)\varepsilon(\boldsymbol{\xi})_{jm}}{\sigma_{jm}^2(i)}, \tag{3.38}$$

$$\boldsymbol{G}_{(n)i} = \sum_{jm\in C_n}\frac{c_{jm}\varepsilon(\boldsymbol{\xi}\boldsymbol{\xi}^T)_{jm}}{\sigma_{jm}^2(i)}, \tag{3.39}$$

$$\varepsilon(\boldsymbol{\xi})_{jm} = [\varepsilon(\boldsymbol{o})_{jm}; 1], \tag{3.40}$$

a

$$\varepsilon(\boldsymbol{\xi}\boldsymbol{\xi}^T)_{jm} = \begin{bmatrix} \varepsilon(\boldsymbol{o}\boldsymbol{o}^T)_{jm} & \varepsilon(\boldsymbol{o})_{jm} \\ \varepsilon(\boldsymbol{o})_{jm}^T & 1 \end{bmatrix}. \tag{3.41}$$

Pro nalezení řešení rovnice (3.37) musíme vyjádřit matici $\boldsymbol{A}_{(n)}$ ve tvaru $\boldsymbol{W}_{(n)}$. Je možné matematicky dokázat, že $\log(|\boldsymbol{A}|) = \log(|\boldsymbol{w}_i^T\boldsymbol{v}_i|)$, kde $\boldsymbol{v}_i$ je kofaktor matice $\boldsymbol{A}_{(n)}$ rozšířený o nulu v poslední dimenzi. Maximalizováním funkce (3.37) dostáváme řešení:

$$\boldsymbol{w}_{(n)i} = \boldsymbol{G}_{(n)i}^{-1}\left(\frac{\boldsymbol{v}_{(n)i}}{f} + \boldsymbol{k}_{(n)i}\right), \tag{3.42}$$

kde $f_{1,2}$ je řešením kvadratické rovnice, jejíž koeficienty jsou

$$[a, b, c] = [\beta_{(n)}, -\boldsymbol{c}_{(n)i}^T\boldsymbol{G}_{(n)i}^{-1}\boldsymbol{k}_{(n)i}, -\boldsymbol{v}_{(n)i}^T\boldsymbol{G}_{(n)i}^{-1}\boldsymbol{v}_{(n)i}], \tag{3.43}$$

$$\beta_{(n)} = \sum_{jm\in C_n}\sum_{t}\gamma_{jm}^{e}(t). \tag{3.44}$$

Po dosazení vypočteného $f_{1,2}$ do rovnice (3.42) dostáváme dvě řešení $\boldsymbol{w}_{(n)i}^{1,2}$. Vybíráme takové, které maximalizuje pomocnou funkci (3.37).

Následně můžeme spočítat logaritmus pravděpodobnosti pro metodou **CMLLR** jako:

$$\log\mathcal{L}(\boldsymbol{o}^{e}(t)|\boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm}, \boldsymbol{A}_{(n)c}, \boldsymbol{b}_{(n)c}) = \log\mathcal{N}(\boldsymbol{o}^{e}(t); \boldsymbol{A}_{(n)c}\boldsymbol{\mu}_{jm} - \boldsymbol{b}_{(n)c}, \boldsymbol{A}_{(n)c}\boldsymbol{C}_{jm}\boldsymbol{A}_{(n)c}^{T}), \tag{3.45}$$

nebo pro metodu **fMLLR** jako:

$$\log \mathcal{L}(\boldsymbol{o}^e(t)|\boldsymbol{\mu_{jm}}, \boldsymbol{C}_{jm}, \boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}) = \log \mathcal{N}(\boldsymbol{A}_{(n)}\boldsymbol{o}^e(t) + \boldsymbol{b}_{(n)}; \boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm}) + 0.5 \log(|\boldsymbol{A}_{(n)}|^2). \tag{3.46}$$

Odhad matice $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$ pomocí (3.42) je iterativní procedura. Naším úkolem je tedy na začátku vhodně inicializovat matice $\boldsymbol{A}_{(n)}$ a $\boldsymbol{b}_{(n)}$. Matice $\boldsymbol{A}_{(n)}$ je obvykle inicializována jako diagonální matice s jednotkovou diagonálou a vektor $\boldsymbol{b}_{(n)}$ je volen jako nulový. Iterace skončí tehdy, když změna v parametrech transformační matice $\boldsymbol{W}_{(n)}$ je zanedbatelná.

3.4.3 Diskriminativní lineární transformace (DLT)

U metody **diskriminativní lineární transformace** (DLT – Discriminative Linear Transformation) je, stejně jako v metodě DMAP v podkapitole 3.3.1, ML kritérium (2.14) nahrazeno některým z diskriminativních kritérií pro trénování (DT), více viz podkapitola 2.3.3.

Diskriminativní MLLR (DMLLR)

V práci [26] je využito pro odvození **diskriminativní MLLR** (DMLLR) DT kritérium MMI (2.32) a tzv. **H-kriteriální funkce** (H-Criterion)

$$(\alpha - 1)\mathcal{F}_{ML}(\lambda) - \mathcal{F}_{MMI}(\lambda), \tag{3.47}$$

kde uživatelsky volitelný parametr $\alpha \geq 1$ zajistí kombinaci kritérií MMI a ML. Kriteriální funkce (3.47) lze dle [26] přepsat do tvaru

$$\begin{aligned} &\sum_{b_{jm} \in C_n} \sum_{e=1}^{E} \sum_{t=1}^{T_e} (\alpha\gamma_{jm}(t) - \gamma_{jm}^{den}(t)) \boldsymbol{C}_{jm}^{-1} \boldsymbol{o}^e(t) \boldsymbol{\xi}_{jm}^T = \\ = &\sum_{b_{jm} \in C_n} \sum_{e=1}^{E} \sum_{t=1}^{T_e} (\alpha\gamma_{jm}(t) - \gamma_{jm}^{den}(t)) \boldsymbol{C}_{jm}^{-1} \boldsymbol{W}(n) \boldsymbol{\xi}_{jm} \boldsymbol{\xi}_{jm}^T, \end{aligned} \tag{3.48}$$

kde $\boldsymbol{\xi}_{jm}$ je rozšířený vektor střední hodnoty j-tého stavu HMM m-té složky GMM, $\boldsymbol{o}(t)$ je vektor pozorování a $\gamma_{jm}(t)$ je aposteriorní pravděpodobnost, že pozorování $\boldsymbol{o}(t)$ je generováno m-tou složkou j-tého stavu HMM. $\gamma_{jm}^{den}(t)$ označuje aposteriorní pravděpodobnost všech přepisů (počítáno dle jmenovatele zlomku (2.32)). Rovnice (3.48) je formálně shodná s rovnicí (3.19) pro výpočet MLLR transformací, jen γ_{jm} je zde nahrazeno $(\alpha\gamma_{jm}(t) - \gamma_{jm}^{den}(t))$. Stejný postup může být aplikován i pro odvození transformací pro kovarianční matice.

Jiný přístup aplikace DT kritéria lze nalézt v [27], kde je využito MPE kritérium na odvození adaptace DMLLR. Změna oproti klasické metodě MLLR spočívá pouze ve změně výpočtu pomocných matic akumulovaných statistik $\boldsymbol{G}$ a $\boldsymbol{k}$ (pozn. je nutno pravděpodobnostní momenty (3.3), (3.4) a (3.5) nahradit rozdílem původní hodnoty a *den* hodnoty vypočtené dle vzorců (3.7), (3.8) a (3.9)):

$$\boldsymbol{k}_{(n)i} = \sum_{b_{jm} \in C_n} \frac{1}{\sigma_{jm}^2(i)} (c_{jm} \boldsymbol{\varepsilon_{jm}}(\boldsymbol{o})(i) + D_{jm} \tilde{\boldsymbol{\mu}}_{jm}(i)) \boldsymbol{\xi}_{jm}, \tag{3.49}$$

$$\boldsymbol{G}_{(n)i} = \sum_{b_{jm} \in C_n} \frac{1}{\sigma_{jm}^2(i)} (c_{jm} + D_{jm}) \boldsymbol{\xi}_{jm} \boldsymbol{\xi}_{jm}^T, \tag{3.50}$$

kde $D_{jm} = fc_{jm}^{den}$ je brzdící faktor pro udržení stability odhadu pomocí diskriminativního kritéria (f je vhodně zvolená brzdící konstanta) a $\tilde{\boldsymbol{\mu}}_{jm}$ je odhadnutá střední hodnota pro m-tou složku j-tého stavu akustického modelu. Střední hodnota $\tilde{\boldsymbol{\mu}}_{jm}$ ze vztahu 3.49 může být spočtena jako:

- střední hodnota z adaptačních dat (pro malý počet dat může být odhad nestabilní)
- odhad střední hodnoty složky použitím MLLR adaptace (znamená větší časovou náročnost)
- odhad střední hodnoty složky použitím MAP adaptace
- původní střední hodnota $\boldsymbol{\mu}_{jm}$ z SI modelu (nezvyšuje časovou náročnost adaptace, pomalejší konvergence metody)

Diskriminativní fMLLR (DfMLLR)

Diskriminativní přístup k metodě **fMLLR** (DfMLLR) popsaný v práci [27] je založen na MMI kritériu. Oproti původní metodě fMLLR se DfMLLR liší opět pouze ve výpočtu pomocných matic $\boldsymbol{G}$ a $\boldsymbol{k}$:

$$\boldsymbol{k}_{(n)i} = \sum_{jm \in C_n} \frac{\boldsymbol{\mu}_{jm}(i)}{\sigma_{jm}^2(i)} (c_{jm} \boldsymbol{\varepsilon}(\boldsymbol{\xi})_{jm} + D_{jm} \boldsymbol{Y}_{jm}), \tag{3.51}$$

$$\boldsymbol{G}_{(n)i} = \sum_{jm \in C_n} \frac{1}{\sigma_{jm}^2(i)} (c_{jm} \boldsymbol{\varepsilon}(\boldsymbol{\xi}\boldsymbol{\xi}^T)_{jm} D_{jm} \boldsymbol{Z}_{jm}, \tag{3.52}$$

kde $\boldsymbol{\varepsilon}(\boldsymbol{\xi})_{jm}$ a $\boldsymbol{\varepsilon}(\boldsymbol{\xi}\boldsymbol{\xi}^T)_{jm}$ je definováno rovnicemi (3.40) a (3.41), $D_{jm} = fc_{jm}^{den}$ je brzdící faktor pro udržení stability odhadu pomocí diskriminativního kritéria (f je vhodně zvolená brzdící konstanta) a

$$\boldsymbol{Z}_{jm} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_{jm} + \tilde{\boldsymbol{\mu}}_{jm} \tilde{\boldsymbol{\mu}}_{jm}^{\mathrm{T}} & \tilde{\boldsymbol{\mu}}_{jm}^{\mathrm{T}} \\ \tilde{\boldsymbol{\mu}}_{jm} & 1 \end{bmatrix}, \tag{3.53}$$

$$\boldsymbol{Y}_{jm} = \begin{bmatrix} \tilde{\boldsymbol{\mu}}_{jm} \\ 1 \end{bmatrix}, \tag{3.54}$$

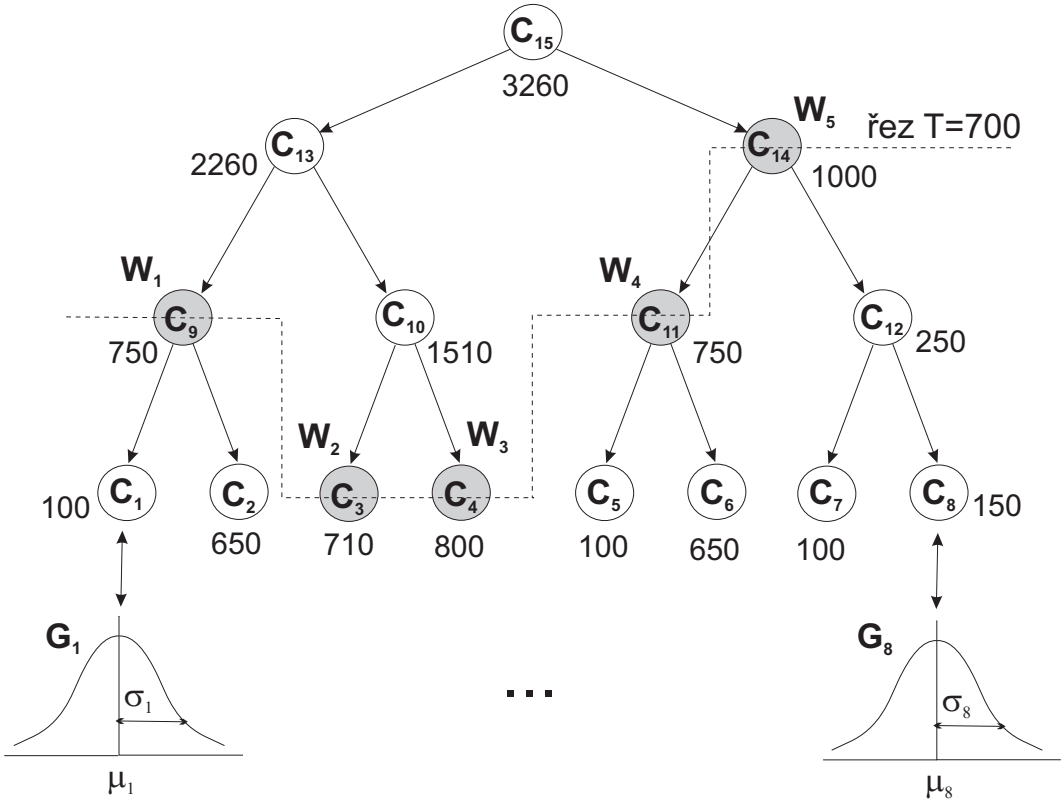
$\tilde{\boldsymbol{\mu}}_{jm}$ a $\tilde{\boldsymbol{\Sigma}}_{jm}$ je odhadnutá střední hodnota a kovarianční matice j-tého stavu a m-té složky, možnosti odhadu jsou popsány v předchozím odstavci pro metodu DMLLR.

Diskriminativní přístupy pro adaptaci založenou na lineárních transformacích ((f)MLLR – fMLLR, MLLR) kvůli akumulaci *den* statistik vyžadují větší počet adaptačních dat a více iterací z důvodu brzdění DT odhadu směrem k ML.

3.4.4 Shlukování podobných parametrů modelu

Výhodou metod založených na lineární transformaci (jako je např. metoda MLLR nebo fMLLR) je možnost nashlukování podobných parametrů modelu (jednotlivé směsi GMM definované střední hodnotou a kovarianční maticí) dle potřeby a množství adaptačních dat. Všechny parametry patřící do jednoho shluku jsou transformovány stejnou transformací. Počet shluků záleží na množství adaptačních dat. Před výběrem shlukovací metody je třeba si položit dvě otázky:

- jak vhodně nashlukovat parametry do jedné třídy, aby pro ně mohla být použita stejná transformace a
- kolik transformací je potřeba pro dané množství adaptačních dat?



Obrázek 3.3: Příklad binárního regresního stromu. C_1 až C_{15} označují jednotlivé uzly, resp. třídy parametrů k nim náležící. Čísla u uzlů značí jejich aktuální obsazení adaptačními daty, šedivě jsou podbarveny ty uzly, které tvoří takzvaný řez stromu, hladinu s dostatečně velkou okupací (větší než práh $Th = 700$). Pro tyto uzly jsou vypočítány transformace $\boldsymbol{W}_1$ až $\boldsymbol{W}_5$. Např. pro třídu C_{12} neexistuje dostatečné množství dat (její okupace adaptačními daty je 250), naopak její rodičovská třída C_{14} má již dostatek pozorování (okupace = 1000) na to, aby pro ni mohla být vypočtena transformace $\boldsymbol{W}_5$. Všechny parametry, které obsahuje třída $C_{14} = C_{11} \cup C_{12}$ jsou použity pro výpočet transformace $\boldsymbol{W}_5$, avšak pouze parametry z třídy C_{12} budou touto transformací adaptovány, protože třída C_{11} má dostatek pozorování pro výpočet své vlastní transformace $\boldsymbol{W}_4$.

Vlastní shluky mohou být vytvořeny a zafixovány před adaptací, pak se jedná o **fixované regresní třídy**. Pro zajištění flexibility a robustnosti shlukování bylo v článku [28] navrženo použití **regresního stromu** pro hierarchické shlukování parametrů modelu do regresních tříd.

Regresní strom (RT - Regresion Tree) je obvykle binárním stromem, kde každý uzel stromu reprezentuje jeden shluk $C_i, i = 1, \ldots, I$, parametrů modelu. Ke každé třídě může být přiřazena transformace $\boldsymbol{W}_{(n)}, n = 1, \ldots, N$, (obvykle je $N < I$, protože se budou počítat pouze ty transformace, pro které je dostatečný počet aktuálních adaptačních dat). Kořenový uzel obsahuje všechny parametry (složky GMM) celého modelu a každý finální list regresního stromu obsahuje pouze jednu konkrétní složku $G_m, m = 1, \ldots, M$, kde v tomto případě M

určuje počet všech komponent všech stavů akustického modelu.

Regresní strom je využit jako apriorní informace o všech možných variantách shlukování v prostoru parametrů modelu. Podle množství a typu adaptačních dat je vybráno vhodné rozdělení prostoru parametrů podle "řezu"(viz příklad na obrázku 3.3) v regresním stromě. Během adaptačního procesu jsou adaptační data rozdělena příslušným Gausovským komponentám (parametrům) modelu a je akumulována tzv. "okupace"(obsazení daty) jednotlivých tříd regresního stromu

$$\beta_{(n)} = \sum_{jm \in C_n} c_{jm} = \sum_{jm \in C_n} \sum_{e=1}^{E} \sum_{t=1}^{Te} \gamma_{jm}^{e}(t), \tag{3.55}$$

kde $\gamma_{jm}^{e}(t)$ je aposteriorní pravděpodobnost, že pozorování $\boldsymbol{o}^{e}(t)$ je generováno m-tou komponentou j-tého stavu HMM, viz rovnice (3.2). Strom je procházen ze zdola nahoru a jsou generovány transformace pouze pro ty uzly stromu, které dosáhnou předem definované úrovně okupace (jejich obsazení daty je větší jak předem definovaný práh Th).

Vytváření regresního stromu

Obvykle dělíme regresní stromy do dvou kategorií, podle informace kterou využívají pro shlukování parametrů.

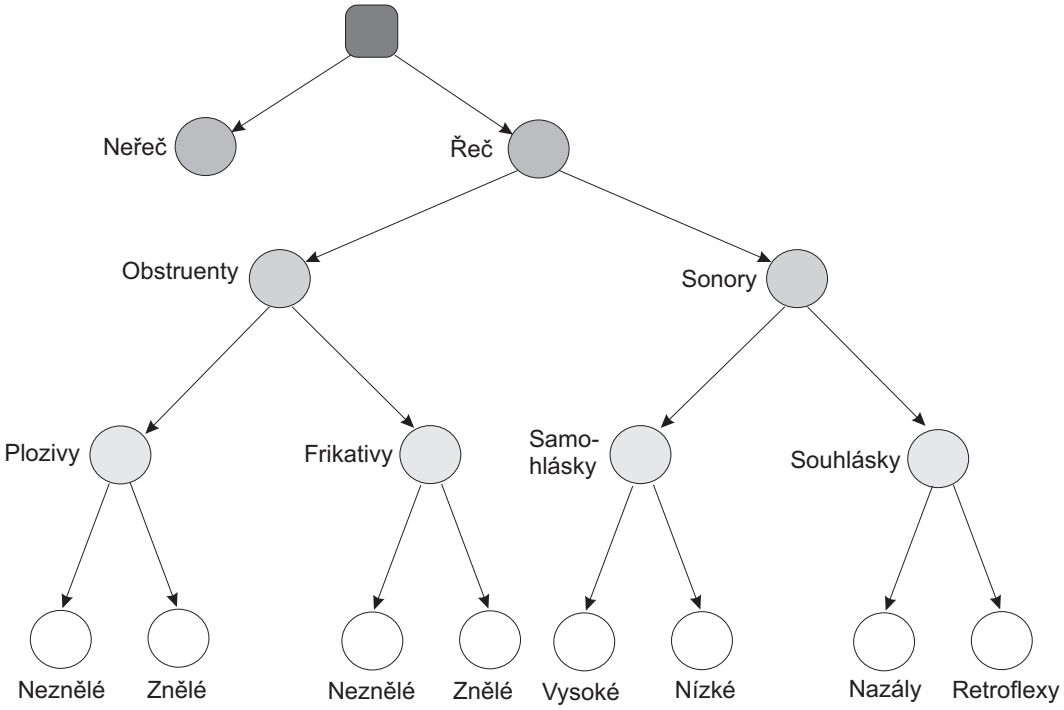
- **Fonetická znalost**. Existují určité expertní znalosti o podobnosti jednotlivých akustických elementů (fonémů), které jsou využity při tvorbě regresního stromu. Příkladem může být fonetický strom na obrázku 3.4, zde jsou akustické jednotky rozděleny do tříd dle fonetického a fonologického hlediska:
  - Souhlásky
    - plozivy (nebo také okluzívy, explozívy)
      - znělé = [b,d,ď,g]
      - neznělé = [p,t,ť,k]
    - frikativy (jde o konstriktivy a emiokluzivy)
      - znělé = [v,z,ž,h,dz,dž]
      - neznělé = [f,s,š,ch,c,č]
    - nazály = [m,n,ň]
    - retroflexy (aproximanty a vibranty) = [l,j,r,ř]
  - Samolásky
    - vysoké = [i,u,í,ú]
    - nízké = [a,e,o,á,é,ó,au,ou]

  Více informací o české fonetické abecedě lze čerpat například v [4].

- **Akustický prostor**. Parametry modelu jsou shlukovány podle vzájemné blízkosti v akustickém prostoru. Tato metoda využívá výhod "data-driven"přístupu, tím nevyžaduje expertní znalost (viz příklad na obrázku 3.3). Dále se v textu budeme věnovat právě tomuto přístupu.

  Optimální rozdělení akustického prostoru na shluky podle [30] vychází z kritéria

$$\hat{Tree} = \arg\max_{Tree} \sum_{s=1}^{S} Q(M, \bar{M}|Tree), \tag{3.56}$$



Obrázek 3.4: Příklad fonetického stromu. Na obrázku je fonetické dělení převzané z [29].

$$Q(M, \bar{M}|Tree) = cost - \frac{1}{2}\log\mathcal{L}(O|M)\sum_{b_{jm}\in M}\sum_{e=1}^{E}\sum_{t=1}^{T_e}K_{jm}(t)[const_{jm}+ \\ +log(|\bar{\boldsymbol{C}}_{jm}|) + (\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})^T\bar{\boldsymbol{C}}_{jm}^{-1}(\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})], \tag{3.57}$$

kde $Tree$ značí regresní strom, M je původní SI model a $\bar{M}$ je nový SA model s parametry $\bar{\boldsymbol{\mu}}_{jm}$ a $\bar{\boldsymbol{C}}_{jm}$. Není však možné garantovat dosažení globálního optima, pouze každé dělení stromu nalezne lokální optimum.

Podle [5] je shlukování prováděno dle středních hodnot a jejich blízkost je dána Eukleidovskou mírou. Konstrukce regresního stromu je prováděna rozdělováním shluků, obvykle se končí v předem definované úrovni stromu. Nepokračuje se tedy až do konečného rozdělení, kde by každému listu odpovídala jedna komponenta GMM.

Jiný přístup shlukování přináší [31], kde tvorba stromu je rozdělena do dvou kroků. V prvním jsou parametry modelu iterativně rozdělovány od vrcholu dolů použitím divizní hierarchické strategie založené na **Bayesově informačním kritériu** (BIC – Bayes Information Criterion) [32], které automaticky odvodí optimální počet finálních tříd. Ve druhém kroku jsou pak finální třídy z prvního kroku iterativně spojovány ze zdola nahoru k vytvoření regresního stromu aglomerativní strategií (blízkost shluků je opět dána BIC kritériem). Výhodou tohoto přístupu je jeho plná automatizace, tedy není při něm potřeba žádné vnější informace (jako je znalost finálního počtu listů).

3.5 Kombinace přístupu MAP a (f)MLLR

Výhodou metody MAP je fakt, že při dostatečném množství dat SA model konverguje k SD modelu. Naopak výhodou metod založených na lineární transormaci je jejich dobrá účinnost i při malém počtu adaptačních dat (možnost shlukování podobných složek hustotních

směsí a tím snižování počtu volných parametrů modelu). Bylo proto navrženo několik postupů kombinujících tyto dva přístupy s předpokládaným využitím výhod z obou metod.

3.5.1 Regresní predikce modelu (RMP)

Při malém množství dat je metoda MAP neúčinná, protože dochází k adaptaci pouze těch parametrů, pro které se vyskytují adaptační data. Z toho důvodu byla do této metody zakomponována myšlenka shlukování podobných parametrů modelu převzatá z metody MLLR. Výsledná metoda se nazývá **regresní predikce modelu** (RMP – Regression-based Model Prediction) [33]. Tato metoda používá malé množství tzv. **zdrojových parametrů**, pro které je dostatečné množství dat, a ty pak využívá k predikci adaptované hodnoty tzv. **cílových parametrů**, které jsou adaptačními daty špatně podmíněné. Pokud předpokládáme lineární vztah mezi zdrojovým parametrem a jeho cílovou skupinou parametrů, lze pak použít lineární regresi k odvození vztahů mezi těmito parametry. Například pro dva parametry x a y lze zapsat lineární vztah

$$y = b_1 x + b_0 + \epsilon, \tag{3.58}$$

kde ϵ označuje chybu aproximace a b_1, b_0 jsou regresní parametry, které lze nalézt např. aplikováním **metody nejmenších čtverců** (LSE – Least Square Error)

$$\arg\min_{b_1,b_0} \sum_{k=1}^{K} \epsilon_k^2 = \arg\min_{b_1,b_0} \sum_{k=1}^{K} (y - b_1 x_k - b_0)^2, \tag{3.59}$$

kde K je konečný počet regresních bodů.

3.5.2 Regrese vážených sousedů (WNR)

Metoda **regrese vážených sousedů** (WNR – Weighted Neighbor Regression) [34] je založena na výše zmíněné technice RMP. Pokud uvažujeme adaptaci pouze středních hodnot $\boldsymbol{\mu}_{jm}$ na novou hodnotu $\bar{\boldsymbol{\mu}}_{jm}$, lze napsat regresní model ve tvaru

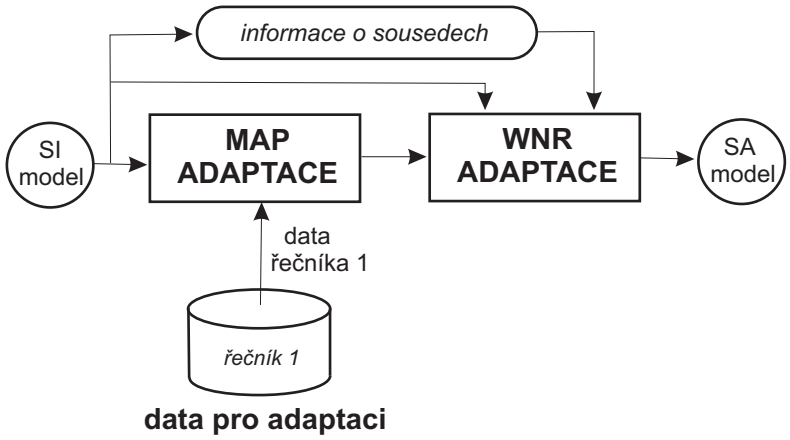
$$\bar{\boldsymbol{\mu}}_{jm} = \boldsymbol{B}\boldsymbol{\mu}_{jm} + \boldsymbol{b}_0 + \boldsymbol{\epsilon}_{jm}. \tag{3.60}$$

Metodou vážených nejmenších čtverců lze nalézt hodnoty regresních transformací minimalizováním vztahu

$$\sum_{k=1}^{K} w_k \boldsymbol{\epsilon}_k^2 = \sum_{k=1}^{K} w_k (\bar{\boldsymbol{\mu}}_k - \boldsymbol{B}\boldsymbol{\mu}_k - \boldsymbol{b}_0)^T (\bar{\boldsymbol{\mu}}_k - \boldsymbol{B}\boldsymbol{\mu}_k - \boldsymbol{b}_0), \tag{3.61}$$

kde všech K středních hodnot $\boldsymbol{\mu}_k$ (parametrů modelu) patří do jedné množiny vzájemně nejbližších (sousedních) středních hodnot $\boldsymbol{\mu}_{jm}$. w_k je váha k-tého parametru v dané množině, která je nepřímo úměrná Mahalanobisově vzdálenosti k-tého parametru od středu množiny.

Postup metody je následovný [35]: Pro každý parametr modelu SI je pomocí Mahalanobisovy vzdálenosti nalezeno K nejbližších parametrů. Po MAP adaptaci jsou všechny komponenty rozděleny podle množství adaptačních dat k nim přidruženým na zdrojové (pro ně bylo k dispozici dostatečné množství dat) a cílové (s malým množstvím adaptačních dat). Pro každý zdrojový parametr a jeho přidruženou množinu sousedů je vypočítána regresní přímka. S její pomocí jsou adaptovány cílové parametry sousedící s daným zdrojovým parametrem (viz obrázek 3.5).



Obrázek 3.5: Blokový diagram WNR adaptace převzatý z [34].

3.5.3 Strukturální MAP (SMAP)

Metoda **strukturální MAP** (SMAP – Structural Maximum A Posteriori) [36] využívá hierarchickou strukturu v prostoru parametrů modelu (jako je regresní strom v podkapitole 3.4.4). Metoda odvozuje transformaci pro každou úroveň této hierarchické struktury. Parametry v konkrétní úrovni jsou použité i pro další své podúrovně. Výsledná transformace parametrů je tedy kombinací transformací vyšších úrovní. Pomocí metody ML lze odhadnout transformace $\boldsymbol{A}_{jm}$ a $\boldsymbol{B}_{jm}$ pro každý uzel binárního dělícího stromu, pak lze střední hodnoty a kovarianční matice transformovat vztahy

$$\bar{\boldsymbol{\mu}}_{jm} = \boldsymbol{\mu}_{jm} + \boldsymbol{B}_{jm}, \tag{3.62}$$

$$\bar{\boldsymbol{C}}_{jm} = \boldsymbol{A}_{jm}\boldsymbol{C}_{jm}. \tag{3.63}$$

Ekvivalentní metody jsou **strukturální MAP s lineární regresí** (SMAPLR – Structural Maximum A Posteriori Linear Regression) lze nalézt v [37] nebo **vážené strukturální MAP** (WSMAP – Weighted Structural Maximum A Posteriori) v [38].

3.5.4 Vyhlazování vektorového pole (VFS)

Metoda **vyhlazování vektorového pole** (VFS – Vector Field Smoothing) [39] transformuje vektory středních hodnot jednotlivých složek stavů akustického modelu, které nebyly adaptovány metodou MAP. Metoda vychází z předpokladu, že akustický prostor jednoho řečníka je spojitě transformovatelný do prostoru jiného řečníka. Po MAP adaptaci, která posune pouze ty parametry modelu, pro které bylo dostatečné množství pozorování v adaptačních datech, metoda VFS transformuje zbylé (špatně podmíněné) vektory středních hodnot. VFS algoritmus prochází třemi kroky[40]:

- výpočet transformačních vektorů pro dobře adaptované vektory středních hodnot metodou MAP, transformační vektor $v_p = \mu_p^R - \mu_p^I$ je dán rozdílem p-té původní μ_p^R a adaptované μ_p^I střední hodnoty modelu, kde $p \in K_1$, (K_1 je množina indexů vektorů středních hodnot, pro které je dostatečné množství adaptačních dat).
- interpolace transformačních vektorů pro neadaptované vektory středních hodnot $v_q = \sum_{k\in N(q)} \lambda_{q,k} v_k / \sum_{k\in N(q)} \lambda_{q,k}$, kde $q \in K_2$, (K_2 je množina indexů vektorů středních

hodnot, pro které nebylo dostatečné množství adaptačních dat), $N(q)$ je množina k-nejbližších vektorů pro vektor μ_q^I a $\lambda_{q,k}$ je váhový koeficient daný blízkostí $d_{q,k}$ vektorů μ_k^I k μ_q^I a je určen vztahem $\lambda_{q,k} = \exp(-d_{q,k}/konst)$. Poté je vektor středních hodnot μ_q^I transformován na nový vektor μ_q^R pomocí vztahu $\mu_q^R = \mu_q^I + v_q$.

- vyhlazení transformačních vektorů $v_p (p \in K_1)$ na vektory v_p^S se provádí kvůli omezení spojité transformovatelnosti, tedy $v_p^S = \sum_{k \in N(q)} \lambda_{,k} v_k / \sum_{k \in N(q)} \lambda_{p,k}$. Výsledný vektor středních hodnot $\mu_p^S = \mu_q^I + v_q^S$ je následně transformován vektorem v_p^S.

Kvůli značné rychlosti byla tato metoda často využívána v on-line adaptaci. Bohužel adaptuje pouze střední hodnoty a kovarianční matice modelu nechává nezměněny, a proto výsledná účinnost metody VFS nepřesáhne účinnost metody MAP adaptující jak střední hodnoty, tak i kovarianční matice a váhové vektory složek stavů akustického modelu.

3.5.5 Maximální aposteriorní pravděpodobnost s lineární regresí ((f)MAPLR)

Metoda **Maximální aposteriorní pravděpodobnost s lineární regresí** (MAPLR – Maximum A Posterior Linear Regression) využívá pro odvození lineárních transformací $\boldsymbol{W}$ apriorní informaci o rozložení transformačních matic $\log p(\boldsymbol{W})$ [41]. Pomocná funkce, odvozená z (3.18) a rozšířená o apriorní informaci, je definována:

$$Q(\lambda, \bar{\lambda}) = \sum_{b_{jm} \in \lambda} \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}(t)(c_{jm} + \log|\bar{\boldsymbol{C}}_{jm}| + (\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})^T \bar{\boldsymbol{C}}_{jm}^{-1}(\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})) + \log p(\boldsymbol{W}). \tag{3.64}$$

Předpokládáme omezení pro transformační matice definované v [41], tedy matice s parametry elipticky symetrického pravděpodobnostního rozložení $\boldsymbol{\mu} = (\boldsymbol{\mu_1}, \ldots, \boldsymbol{\mu}_d)^T$ a $\Delta = diag(\boldsymbol{\Sigma_1}, \ldots, \boldsymbol{\Sigma_d})$. Postupujeme-li podle MLLR notace (viz podkapitola 3.4.1), lze řádek transformační matice odvozené z MAPLR vypočítat jako:

$$\boldsymbol{w}_i^T = \bar{\boldsymbol{z}}_i^T \bar{\boldsymbol{G}}_i^{-1}, \tag{3.65}$$

kde

$$\bar{\boldsymbol{G}}_i = \boldsymbol{G}_i + \boldsymbol{\Sigma}_i^{-1}, \tag{3.66}$$

a

$$\bar{\boldsymbol{z}}_i = \boldsymbol{z}_i + \boldsymbol{\mu}_s^i \boldsymbol{\Sigma}_i^{-1}. \tag{3.67}$$

Při uvažování apriorní informace dosahuje lineární regrese lepších výsledků, než samotná MLLR metoda. MAPLR využívá z obou kombinovaných metod to nejlepší. Tedy jako MAP využívá apriorní informaci a z MLLR pak vlastnost adaptovat i parametry modelu, která nejsou v adaptačních datech zastoupena pozorováním.

Ekvivalentní postup pro kombinaci fMLLR a MAP, tedy metoda **Maximální aposteriorní pravděpodobnost s lineární regresí pro vektory pozorovaní** (fMAPLR – feature Maximum A Posterior Linear Regression) je popsaná v práci [42].

3.6 Shlukování mluvčích (SC)

Adaptační strategie popsaná v této části, metoda **shlukování mluvčích** (SC – Speaker Clustering) [43], je založena na hledání podmnožiny řečníků z trénovací množiny, kteří jsou

akusticky blízko k rozpoznávanému řečníku. K přepočítání parametrů modelu jsou s výhodou použita data od nejbližších řečníků (apriorní znalost), než celá kompletní trénovací databáze obsahující promluvy od velkého množství řečníků. Nový model má pak mnohem blíže k rozpoznávaným datům než původní SI model. Jednou z možných implementací tohoto přístupu je na pohlaví závislý model (GD – Gender Dependent).

Adaptace na řečníka probíhá v těchto krocích:

- 1. Vytvoření akustického modelu z celé trénovací databáze (SI model).
- 2. Vytvoření akustických modelů pro všechny řečníky vyskytující se v trénovací databázi. Pokud nemáme dostatečné množství dat pro natrénování SD modelu, lze použít některou z adaptačních metod ke konstrukci SA modelu, popřípadě natrénovat pouze jednostavový GMM a použít některou z metod identifikace řečníka.
- 3. Pro adaptační data od řečníka, jehož řeč se rozpoznává, nalézt N nejbližších řečníků (výběr nejpravděpodobnějších modelů pro adaptační data). K rychlejšímu výběru nejlepších řečníků může být použit i regresní strom viz [44].
- 4. Z trénovacích dat od nejbližších řečníků vytvořit adaptovaný model. Obvykle se adaptují jen střední hodnoty, přičemž zbylé parametry zůstávají shodné s SI modelem. Jednou z možností vytvoření nového modelu je kombinace vektorů středních hodnot nejbližších řečníků metodou MAP či ML [45].

Metoda SC si vystačí s malým množstvím adaptačních dat, jejím cílem je najít si podobná data v trénovací množině (data od nejbližších řečníků) k rozpoznávanému řečníkovi. Výhodou také je, že adaptace modifikuje všechny parametry SI modelu, nejen ty, které byly obsazeny adaptačními daty rozpoznávaného řečníka.

Kapitola 4

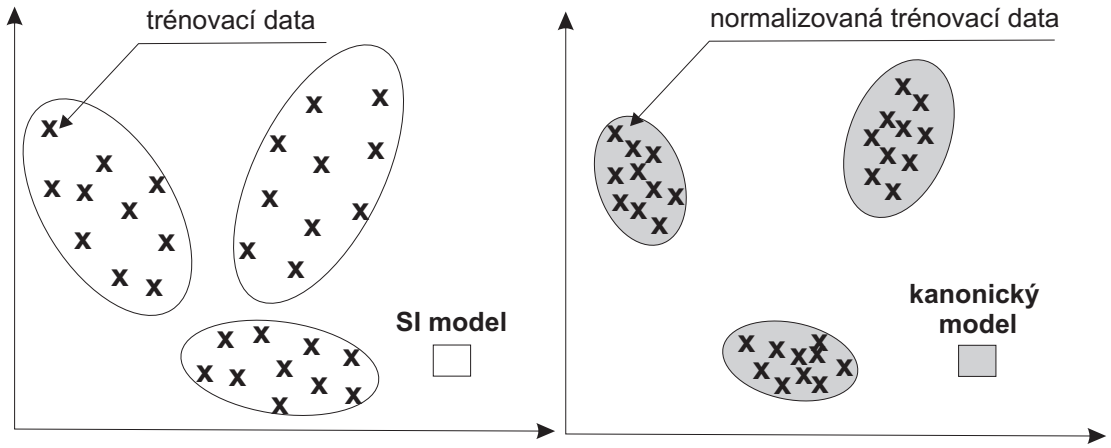
Adaptační techniky pro trénování

Tato kapitola popisuje adaptační techniky pro využití v trénovací fázi. Namísto adaptace SI modelu pomocí transformací vypočtených z dat dostupných v adaptační fázi (jak bylo popsáno v kapitole 3) jsou tyto adaptační metody aplikovány na trénovací data, z kterých je pak vytvořen model bez rušivé informace o řečníkovi.

Postup je založen na hypotéze, že variabilita v akustickém modelu SI je způsobena jak fonetickou odlišností jednotlivých subslovních elementů (tuto informaci využíváme), tak rozdílem hlasových charakteristik mluvčích z trénovací databáze (které jsou pro rozpoznávání řeči rušivé) a vlivů prostředí, ve kterém byla trénovací data nahrána (aditivní či konvolutorní šum). Výsledkem je větší variabilita v trénovacích datech u SI modelu, než u SD modelu. Cílem adaptačních technik používaných při trénování modelu je odstranění právě této na řečníkovi a prostředí závislé nežádoucí informace. Metody se snaží snížit rozptyl trénovacích dat pro konkrétní subslovní jednotku a tím zajistit její lepší separovatelnost od ostatních jednotek. Na rozdíl od předchozí kapitoly 3, kdy byly charakteristiky modelu přizpůsobeny konkrétnímu řečníku, je zde vytvářen tzv. **kanonický model**, z něhož je informace o řečníkovi pomocí adaptačních technik odstraněna (viz obrázek 4.1).

Kanonický model reprezentuje veškerou požadovanou řečovou variabilitu celé trénovací databáze, ale je nezávislý na akustických podmínkách. Takovýto model je vytvářen jen tehdy, když máme k dispozici množinu transformací pro odstranění neřečové variability v datech. Tvar kanonického modelu závisí na formách adaptačních transformací. Pro lineární transformace jde o standardní HMM. Kanonický model je mnohem více kompaktní, je tedy nutné jej při vlastním rozpoznávání dále adaptovat na konkrétní testované akustické podmínky.

Možností, jak snížit variabilitu v trénovacích datech, je hned několik. Za zmínění stojí například **kepstrální normalizace** (CMN – Cepstrum Mean Normalization) [46], která je jednoduchou a často používanou metodou k odstranění vlivu kanálu. Další z metod je tzv. **gaussionalizace** (*Gaussianisation*) viz [47], normalizující kumulativní hustotní funkci vektorů pozorování na standardní Gaussovské rozložení. Sofistikovanější přístupy [23] jsou **trénování s adaptací na mluvčího** (SAT – Speaker Adaptive Training) [48], [49], **trénování s adaptací pomocí shlukování mluvčích** (CAT – Cluster Adaptive Training) [50] a **normalizace délky hlasového traktu** (VTLN – Vocal Tract Length Normalization) [51], [52]. Výhodou zmíněných metod je, že se dají snadno a úspěšně kombinovat dohromady.



Obrázek 4.1: Ilustrativní příklad rozdílné variability složek modelu SI a kanonického modelu.

4.1 Trénování s adaptací na mluvčího (SAT)

Metoda **trénování s adaptací na mluvčího** (SAT – Speaker Adaptive Training) využívá lineárních transformací popsaných v podkapitole 3.4. Metoda se snaží odstranit variabilitu řečníků z fonetické informace a vytvořit kompaktní kanonický model λ_C, který informaci o řečníkovi neobsahuje. Zatímco klasická adaptace hledá model $\hat{\lambda}$, který by maximalizoval pravděpodobnost adaptačních dat od všech řečníků S

$$\hat{\lambda} = \arg\max_{\lambda} \prod_{s=1}^{S} P(\boldsymbol{O}^s|\lambda), \tag{4.1}$$

SAT počítá na řečníku s závislou transformaci H^s ke kanonickému modelu λ_C tak, aby se maximalizovala pravděpodobnost [25]

$$(\hat{\lambda}_C, \hat{\boldsymbol{H}}) = \arg\max_{(\lambda_C, \boldsymbol{H})} \prod_{s=1}^{S} P(\boldsymbol{O}^s|\boldsymbol{H}^s(\lambda_C)), \tag{4.2}$$

tedy hledáme kanonický model $\hat{\lambda}_C$ a jeho transformaci $\hat{\boldsymbol{H}}^s$ závislou na řečníkovi s, které budou maximalizovat pravděpodobnost pro každého řečníka s zvlášť. Fonetická informace je uložena v kanonickém modelu $\hat{\lambda}_C$, informace od řečníka pak v transformaci $\hat{\boldsymbol{H}}^s$. Kanonický model, spolu s některou z adaptačních metod (viz předešlá kapitola 3) použitou při fázi rozpoznávání, zajistí výsledky lepší, než lze získat s původním SI modelem.

4.1.1 SAT pro MLLR

Při klasickém trénování akustického modelu se využívá EM algoritmus (viz podkapitola 2.3.1), který se snaží maximalizovat pomocnou funkci (2.20) vedoucí k odvození parametrů modelu, které zvyšují pravděpodobnost pro trénovací data.

V SAT přístupu [48] je naší snahou maximalizovat pomocnou funkci

$$Q(\rho, \hat{\rho}) = \sum_{s} \sum_{t} \sum_{jm} \gamma_{jm}^{s}(t) \mathcal{N}(\boldsymbol{o}^s(t); \hat{\boldsymbol{A}}^s \hat{\mu}_{jm} + \hat{\boldsymbol{b}}^s, \hat{\boldsymbol{C}}_{jm}), \tag{4.3}$$

kde parametr $\rho = (\boldsymbol{H}^s, \lambda_C) = ((\boldsymbol{A}^s, \boldsymbol{b}^s), (\hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{C}}_{jm}))$ se skládá z transformace na řečníka a z kanonického modelu.

Pro zjednodušení výpočtu je maximalizace rozdělena iterativně na tři části. V každé z nich se snažíme optimalizovat pouze jeden z parametrů, zatímco zbylé dva zůstávají fixovány. V každé části optimalizačního procesu nesmí hodnota pomocné funkce Q klesat:

$$Q(\boldsymbol{H}^s, \lambda_C) \leq Q(\hat{\boldsymbol{H}}^s, \lambda_C) \leq Q(\hat{\boldsymbol{H}}^s, (\hat{\boldsymbol{\mu}}, \boldsymbol{C})) \leq Q(\hat{\boldsymbol{H}}^s, \hat{\lambda}_C). \tag{4.4}$$

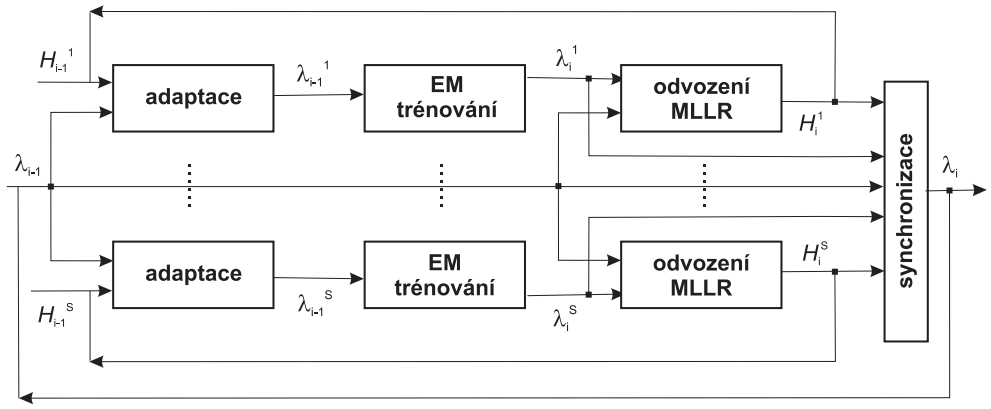
Konkrétně rovnice pro střední hodnoty a kovarianční matice kompaktního modelu lze zapsat ve formě

$$\hat{\boldsymbol{\mu}}_{jm} = \left(\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t) \hat{\boldsymbol{A}}^{sT} \hat{\boldsymbol{C}}_{jm}^{-1} \hat{\boldsymbol{A}}^s \right)^{-1} \sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t) \hat{\boldsymbol{A}}^{sT} \hat{\boldsymbol{C}}_{jm}^{-1} (\boldsymbol{o}^s(t) - \boldsymbol{b}^s), \tag{4.5}$$

$$\hat{\boldsymbol{C}}_{jm} = \frac{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t) (\boldsymbol{o}^s(t) - \hat{\boldsymbol{\mu}}_{jm}^s)(\boldsymbol{o}^s(t) - \hat{\boldsymbol{\mu}}_{jm}^s)^T}{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t)}, \tag{4.6}$$

kde odhad transformace $\boldsymbol{H}^s$ je proveden pomocí standardní metody MLLRmean (viz podkapitola 3.4.1) a $\hat{\boldsymbol{\mu}}_{jm}^s = \boldsymbol{A}^s \hat{\boldsymbol{\mu}}_{jm} + \boldsymbol{b}^s$ je transformovaná střední hodnota kanonického modelu.

Re-estimační SAT proces je zobrazen na obrázku 4.2, kde celková zpětná vazba značí, že proces může být opakován, dokud model nedokonverguje do svého optima.



Obrázek 4.2: Blokový diagram pro metodu SAT založenou na MLLR transformacích. První blok zadaptuje model pomocí transformací $\boldsymbol{H_{i-1}}$, druhý blok odvodí nové parametry modelu λ_i (viz rovnice (4.5) a (4.6)), třetí blok pak spočte nové transformační matice $\boldsymbol{H_i}$ pomocí klasických adaptačních metod. Celý proces lze iterativně opakovat. Obrázek je převzat z [49].

Nevýhodou tohoto přístupu k trénování je značná paměťová náročnost [49], protože je potřeba uchovávat v paměti každou střední hodnotu a kovarianční matici kanonického modelu spolu s transformací, a to pro každého řečníka s zvlášť. S tím je též spojena časová náročnost díky I/O operacím při práci s pamětí. Redukování náročnosti metody je navrženo v [49].

4.1.2 SAT pro fMLLR

Druhý přístup k SAT navržený v [23] je založen na metodě fMLLR (viz podkapitola 3.4.2). Jeho výhodou oproti předchozí metodě je, že adaptační transformace jsou počítány pro trénovací vektory pozorování. Tím je značně ušetřen čas a paměť pro ukládání mezivýsledků, protože přepočet středních hodnot a kovariančních matic probíhá v jednom optimalizačním kroku právě z výsledných transformovaných vektorů pozorování.

Pomocná funkce má tvar

$$Q(\rho, \hat{\rho}) = \sum_s \sum_t \sum_{jm} \gamma_{jm}^s(t) \mathcal{N}(\hat{\boldsymbol{o}}^s(t); \boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm}). \tag{4.7}$$

Dosadíme-li do této rovnice (4.7) vztah pro transformovaný vektor pozorování $\hat{\boldsymbol{o}}^s(t) = \boldsymbol{A}^s \boldsymbol{o}^s(t) + \boldsymbol{b}^s$, s využitím rovnice (3.46) dostáváme

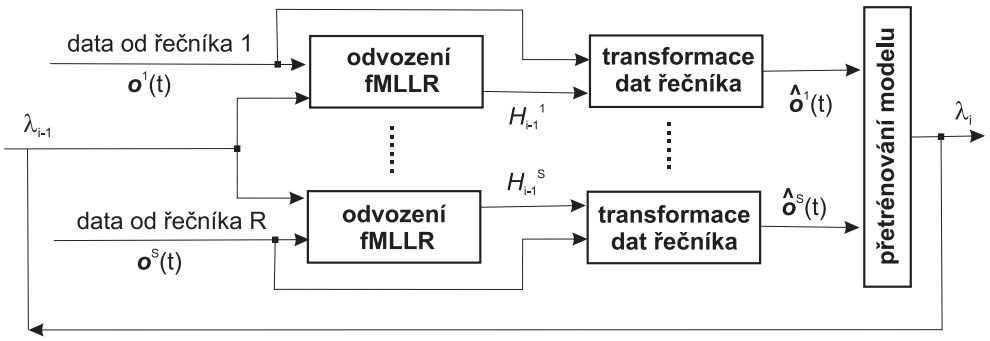
$$\begin{aligned} Q(\rho, \hat{\rho}) = c - \frac{1}{2} \sum_s \sum_t \sum_{jm} \gamma_{jm}^s(t)(c_{jm} + \\ + \log(|\hat{\boldsymbol{C}}_{jm}|) - log(|\boldsymbol{A}^{s2}|) + (\hat{\boldsymbol{o}}^s(t) - \hat{\boldsymbol{\mu}}_{jm})^T \hat{\boldsymbol{C}}_{jm}^{-1} (\hat{\boldsymbol{o}}^s(t) - \hat{\boldsymbol{\mu}}_{jm})). \end{aligned} \tag{4.8}$$

Transformační matice $\boldsymbol{H}^s = (\boldsymbol{A}^s, \boldsymbol{b}^s)$ jsou odvozeny adaptační metodou fMLLR (viz podkapitola 3.4.2) pro dané trénovací vektory pozorování od konkrétního řečníka. Střední hodnoty a kovarianční matice kanonického modelu lze poté přepočítat s využitím znalosti o transformacích vektorů pozorování v jednom kroku

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t) \hat{\boldsymbol{o}}^s(t)}{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t)}, \tag{4.9}$$

$$\hat{\boldsymbol{C}}_{jm} = \frac{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t) (\boldsymbol{o}^s(t) - \hat{\boldsymbol{\mu}}_{jm}^s)(\boldsymbol{o}^s(t) - \hat{\boldsymbol{\mu}}_{jm}^s)^T}{\sum_{s=1}^{S} \sum_{t=1}^{T_s} \gamma_{jm}^s(t)}, \tag{4.10}$$

Stejně jako u předchozí uvedené metody 4.1.1, i tento postup lze iterativně opakovat (viz obrázek 4.3).



Obrázek 4.3: Blokový diagram pro metodu SAT založenou na fMLLR transformacích. Nejprve se odvodí transformační matice $\boldsymbol{H_{i-1}}$ metodou fMLLR, kterými se zadaptuje vektor příznaků $\boldsymbol{o(t)}$. Z nových příznaků $\hat{\boldsymbol{o}}(\boldsymbol{t})$ se přetrénuje model λ_{i-1} (pomocí vztahů (4.9) a (4.10)). Postup lze iterativně opakovat.

4.1.3 Diskriminativní adaptace pro trénování (DAT)

Metoda **diskriminativní adaptace pro trénování** (DAT – Discriminative Adaptation Training) je diskriminativní verzí SAT. Například v [53] je odvozena metoda adaptačního trénování vycházející z **diskriminativní lineární transformace pro vektory pozorování** popsané v kapitole 3.4.3 a MMI kritéria (2.32). Jiný přístup, uvedený v [54], používá kritérium MPE (2.33).

V DAT, stejně jako v metodě SAT, je každá iterace rozdělena do dvou kroků, nejprve se odhadnou lineární transformace a poté parametry kanonického modelu. Diskriminativní kritérium (MMI popř. MPE) je používáno v obou krocích. Opět je s výhodou využívána omezená transformace vektorů pozorování, spíše než neomezená transformace vyžadující značné paměťové nároky.

4.2 Trénování s adaptací pomocí shlukování mluvčích (CAT)

Metoda **trénování s adaptací pomocí shlukování mluvčích** (CAT – cluster Adaptive Training) je jednoduchým rozšířením metody **shlukování mluvčích** (viz podkapitola 3.6). Poznamenejme, že máme trénovací data všech řečníků rozdělená do P shluků dle akustické blízkosti. Nad každým shlukem je vytvořen model (ať již trénováním nebo adaptací SI modelu). Množina $\mathcal{M}$ (4.14) těchto shlukových modelů nahrazuje jeden kanonický model používaný v metodě SAT (viz podkalpitola 4.1).

K vytvoření akustického modelu pomocí metody CAT [11] je využit vektor interpolačních vah $\boldsymbol{\nu}^s$ pro kombinaci všech středních hodnot P shlukových modelů, obvykle sdružených do matice

$$\boldsymbol{M}_{jm} = [\mu_{jm}^1, \ldots, \mu_{jm}^P] \text{ pro } jm = 1, \ldots, M \ , \tag{4.11}$$

kde M je celkový počet všech složek všech stavů modelu, P je počet shluků. Metoda CAT se zaměřuje na adaptaci pouze středních hodnot akustického modelu $\boldsymbol{\mu}$, zbylé parametry shlukových kanonických modelů (pravděpodobnosti přechodů a a kovarianční matice $\boldsymbol{C}$) zůstávají nezměněny.

Vektor vah

$$\boldsymbol{\nu}^s = [\nu^{1r}, \ldots, \nu^{Ps}], \tag{4.12}$$

hrající úlohu transformace, je počítán pro každé odlišné akustické podmínky $s = 1, \ldots, S$ (různý řečník či různé prostředí).

Adaptovaná střední hodnota j-tého stavu m-té komponenty pro jednotlivé akustické podmínky s je dána vztahem

$$\boldsymbol{\mu}_{jm}^s = \boldsymbol{M}_{jm}\boldsymbol{\nu}^s. \tag{4.13}$$

4.2.1 Hledání parametrů modelu a transformací

Stejně jako v metodě SAT, i zde se pro trénování používá ML kritérium [50]. Změna je pouze v kanonickém modelu, který je zde tvořen množinou středních hodnot jednotlivých shlukových modelů a kovariančních matic, které mají všechny shlukové modely stejné

$$\mathcal{M} = \{\{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_M\}, \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M\}\}. \tag{4.14}$$

Pro odvození parametrů kanonického modelu $\mathcal{M}$ a váhových vektorů (transformací) $\boldsymbol{\Upsilon} = \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^S$ se s výhodou používá EM algoritmus (viz podkapitola 2.3.1). Pomocná funkce pak má tvar

$$Q(\mathcal{M}, \boldsymbol{\Upsilon}, \hat{\mathcal{M}}, \hat{\boldsymbol{\Upsilon}}) = -\frac{1}{2}\sum_s \sum_{jm} \sum_t \gamma_{jm}(t) \left((\boldsymbol{o}^s(t) - \boldsymbol{M}_{jm}\nu^s)^T C_{jm}^{-1} (\boldsymbol{o}^s(t) - \boldsymbol{M}_{jm}\nu^s) \right), \tag{4.15}$$

kde $\mathcal{M}$ je starý kanonický model a $\hat{\mathcal{M}}$ je nově odvozený model (analogicky i pro $\boldsymbol{\Upsilon}$). Je nesnadné odvozovat kanonický model $\mathcal{M}$ a transformace $\boldsymbol{\Upsilon}$ společně, proto se odhad provádí ve dvou krocích, nejprve $\hat{\boldsymbol{\Upsilon}}$ a pak $\hat{\mathcal{M}}$. Obvykle se postup iterativně opakuje dokud kritérium nezačne konvergovat.

4.2.2 Reprezentace shluků

Existují dvě možnosti jak reprezentovat střední hodnoty jednotlivých shluků, **CAT založené na modelu** a **CAT založené na transformacích**. Jejich kompletní popis je uveden v [50]. V prvním zmíněném způsobu je každý shluk přímo reprezentován akustickým modelem, druhý způsob popisuje shluk adaptační maticí, která transformuje globální model na model daného shluku (např. metoda MLLRmean, viz podkapitola 3.4.1). Pro inicializaci kanonických modelů [11] se s výhodou využívají **dekompozice vlastních hlasů** (ED – Eigenvoices Decomposition) (více viz podkapitola 6.4).

Výhodou metody CAT je rychlá adaptace pro malý objem adaptačních dat. V porovnání s jinými adaptačními metodami, jako je např. SAT, metoda CAT vyžaduje znatelně menší počet adaptačních parametrů (dimenze P váhového vektoru je obvykle v jednotkách). Čím méně parametrů je pro metodu CAT použito, tím menší je její efektivita v porovnání s metodou SAT. Metoda může být také snadno a efektivně rozšířena jinou adaptací viz [55].

4.2.3 Diskriminativní adaptace pro trénování pomocí shlukování (DCAT)

Rozšířením přístupu CAT je **diskriminativní adaptace pro trénování pomocí shlukování mluvčích** (DCAT – Discriminative CAT). Na rozdíl od klasické metody CAT, jednotlivé shlukové modely jsou zde trénovány pomocí disktiminativních metod z podkapitoly 2.3.3. Ty však vyžadují mnohem více dat, než je třeba pro klasické trénování založené na ML kritériu (2.14). Navržené metody využívající diskriminativní kritéria MMI (2.32) či MPE (2.33) jsou uvedeny v [56].

4.3 Normalizace délky hlasového traktu (VTLN)

Důvodů řečové variability mezi řečníky je velké množství, např. lingvistické odlišnosti, způsob artikulace, zdravotní a psychický stav řečníka a jiné. Převládajícím faktorem však je rozlišná fyziologická stavba hlasového ústrojí. Jedním z hlavních zdrojů odlišnosti řečníků je rozdílná délka hlasové trubice, která se může pohybovat od 13 cm pro ženy do 18 cm pro muže. Délka hlasového traktu zásadně ovlivňuje polohu formantových frekvencí (a to s nepřímou úměrou), které jsou detekovány převážně u znělých hlásek.

Metoda **normalizace délky hlasového traktu** (VTLN – Vocal Tract Length Normalization) [51] se snaží kompenzovat projevy různé délky hlasové trubice v řeči transformováním frekvenční osy řečníka tak, aby se jeho pozice formantů blížily pozicím průměrného řečníka.

4.3.1 Transformační funkce

Transformace frekvenční osy spočívá v jejím nelineárním natažení (popř. smrštění), odborně nazývaném **borcení** (*warping*). Warpovacích funkcí $\tilde{\omega} = \mathcal{F}_\alpha(\omega)$ je celá řada, nejpoužívanější z nich jsou podle [57] tyto:

- 1. Po částech lineární funkce

$$\mathcal{F}_\alpha(\omega) = \begin{cases} \alpha\omega & \text{pro } 0 \leq \omega < \omega_0 \\ \alpha\omega + \frac{\pi - \alpha\omega_0}{\pi - \omega_0}(\omega - \omega_0) & \text{pro } \omega_0 \leq \omega \leq \omega_m \end{cases}, \tag{4.16}$$

  kde frekvence ω_0 je rovna nebo větší jak průměrná frekvence třetího formantu a α je warpovací faktor. Průběh funkce je zobrazen na obrázku 4.4a.

- 2. Bilineární funkce

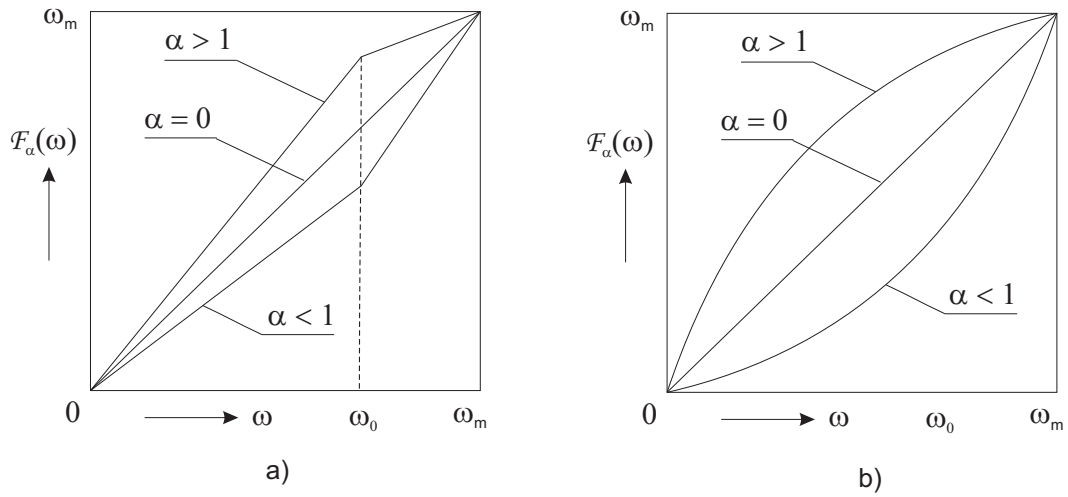
$$\mathcal{F}_\alpha(\omega) = \omega + 2\arctan\left(\frac{(1-\alpha)\sin\omega}{1-(1-\alpha)\sin\omega}\right), \tag{4.17}$$

  Průběh funkce je zobrazen na obrázku 4.4b.

- 3. *Sine-log all-pass* transformace (SLAPT)

$$\mathcal{F}_\alpha(\omega) = \omega + \sum_{k=1}^{K} \alpha_k \sin(\pi k\omega). \tag{4.18}$$

  Tato funkce byla představena v práci [58] a je vhodná pro adaptaci s K třídami.



Obrázek 4.4: Warpovací funkce a) po částech lineární, b) bilineární.

Pro tyto funkce platí, že původní frekvenční osa je transformována na stejný interval $\tilde{\omega} \in \langle 0, \omega_m \rangle$ a má dva fixované body $\mathcal{F}_\alpha(0) = 0$ a $\mathcal{F}_\alpha(\omega_m) = \omega_m$. Naším úkolem je nalézt pro každého řečníka jeho warpovací faktor α tak, aby byl nejlépe znormalizován jeho hlasový trakt. α je obvykle hledáno z intervalu $\langle 0,88; 1,12 \rangle$ [4]. Jinou warpovací funkci s více než jedním proměnným parametrem lze nalézt např. v [59].

Zatímco metoda SAT odvozená z MLLR přístupu popsaném v podkapitole 4.1.1 je zaměřena na transformaci parametrů modelu, metoda VTLN pracuje s vektory příznaků (stejně jako SAT na fMLLR z podkapitoly 4.1.2). Znormované vektory pomocí warpovací funkce však již nelze rozpoznat aktuálním modelem, ale je nutno akustický model znovu natrénovat na warpovaných nahrávkách (proto je VTLN metoda zařazena v této kapitole). Předpokládáme-li, že vektor pozorování je parametrizován pomocí metody **melovských frekvenčních kepstrálních koeficientů** (MFCC – Mel Frequency Cepstral Coefficient), popř. metodou **perceptivní lineární prediktivní analýzy** (PLP – Perceptual Linear Predictive), lze warpování frekvenční osy provádět buď přímo přes spektrální vzorky nebo transformovat meze jednotlivých pásem v bance filtrů. Druhý ze zmíněných způsobů je výpočetně méně náročný.

4.3.2 Odhad warpovacího faktoru

Nalezení optimálního warpovacího faktoru s-tého řečníka α^{s*} jde ruku v ruce s optimalizačním kritériem pro rozpoznávání. Označme soubor trénovacích promluv s-tého řečníka

$\boldsymbol{O}^s = \{\boldsymbol{O}^{1s}, \ldots, \boldsymbol{O}^{Es}\}$ a k němu odpovídající soubor přepisů $\boldsymbol{W}^s = \{\boldsymbol{W}^{1s}, \ldots, \boldsymbol{W}^{Es}\}$, pak soubor těchto promluv warpovaných faktorem α můžeme označit $\boldsymbol{O}^s_\alpha = \{\boldsymbol{O}^{1s}_\alpha, \ldots, \boldsymbol{O}^{Es}_\alpha\}$.

Optimální warpovací faktor pro daného řečníka lze nalézt maximalizací věrohodnosti warpovaných promluv za předpokladu SI modelu λ a daného přepisu promluv $\boldsymbol{W}^s$

$$\alpha^{s*} = \arg\max_\alpha P(\boldsymbol{O}^s_\alpha|\lambda, \boldsymbol{W}^s). \tag{4.19}$$

Pro zjednodušené hledání warpovacího faktoru α^{s*} byl navržen vhodný interval doporučených hodnot α, uvedený v této kapitole 4.3.1.

Jiné, než výše uvedené ML kritérium pro výběr optimálního warpovacího faktoru, tzv. **lineární diskriminativní kritérium** (LD – Linear Discriminant), lze nalézt v [60]. Je založeno na kovariančních maticích daných akustických vzorků. Předpokládáme, že každý vzorek je přidružen do některé z akustických tříd. Pak LD kritérium má formu

$$LD = \frac{|T|}{|W|}, \tag{4.20}$$

kde T je kovarianční matice všech vzorků a W je průměrná kovarianční matice vzorků patřících do konkrétních tříd c_i

$$W = \sum_i p(c_i) W_i. \tag{4.21}$$

Hledáme takový warpovací parametr α^*_i, který bude maximalizovat kritérium (4.20). Tehdy budou různé třídy vzorků od sebe vzájemně daleko, ale mají v průměru malý rozptyl mezi svými vzorky. Tato metoda je také použita pro rychlou transformaci při on-line použití v [57].

4.3.3 Normalizovaný akustický model

S pomocí warpovaných promluv $\boldsymbol{O}^s_\alpha$ lze natrénovat kompaktní model λ_c, který je "na míru ušitý"na řečníka s průměrnou délkou vokálního traktu. Při procesu rozpoznávání je pak nutné testované promluvy normalizovat příslušným warpovacím faktorem. VTLN pro unsupervised adaptaci je popsáno v práci [61].

4.4 Normalizace délky hlasového traktu pomocí lineárních transformací (VTLN-LT)

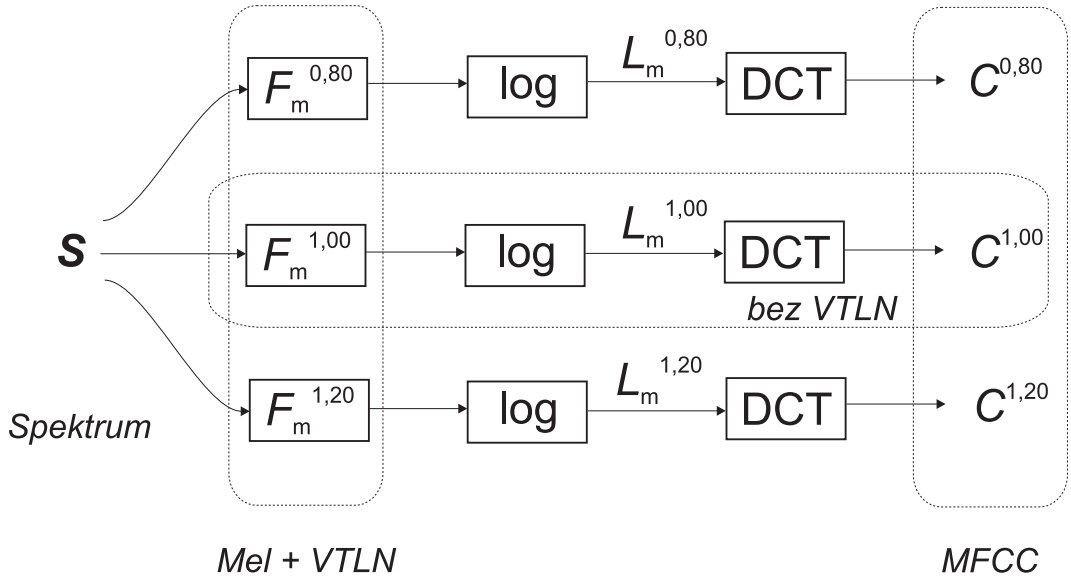
Zatímco v předchozí kapitole 4.3.2 byl odhad warpovacího faktoru otázkou neustálého parametrizování vstupní promluvy kvůli hledání maxima pravděpodobnosti (viz obrázek 4.5), např. v [62] a [63] je popsán postup, jak tomuto zdlouhavému procesu ulevit.

Warpovací faktor může být efektivně odvozen z akumulovaných statistik (viz kapitola 3.2), protože proces warpování je možné obejít lineární transformací [64]. VTLN proces zobrazený na obrázku 4.5 [65] popisuje, jak je S (spektrum signálu) filtrováno bankou filtrů (vyhlazenou jak Mel fitrací - F_m, tak případně i VTLN warpovacím faktorem - F^α_m) a po zlogaritmovaní (blok log) provedena **diskrétní kosinová transformace** (DCT – Discrete Cosine Transformation). Pro výsledné newarpované cepstrum $C^{1,00}$ (MFCC – Mel Frequency Cepstral Coefficient) platí

$$C^{1,00} = \mathrm{DCT}[\log(F_m S)], \tag{4.22}$$

a warpované cepstrum C^α je dáno

$$C^\alpha = \mathrm{DCT}[\log(F^\alpha_m S)]. \tag{4.23}$$



Obrázek 4.5: Schéma výpočtu parametrizace MFCC normalizované pomocí VTLN pro hodnoty warpovacího faktoru 0,80; 1,00 a 1,20.

4.4.1 Odvození lineárních transformací

Vztah mezi $C^{1,00}$ a C^{α}:

$$C^{\alpha} = \mathrm{DCT}[\log(F_m^{\alpha}\{F_m^{-1} \exp \mathrm{DCT}^{-1}(C^{1,00})\})] \tag{4.24}$$

nelze považovat za lineární transformaci, protože všechny průběžné operátory nejsou lineární (kvůli funkci log) a pro operátor F_m^{-1} nemusí být v praxi zaručena jeho existence, tedy spektrum S je nemožno zpětně odvodit z kepstra C.

Řešením je separovat VTLN od Mel filtrace. Nechť $L_m = \log(F_m S)$ je výstup z banky Mel-filtrů a $L_m^{\alpha} = \log(F_m^{\alpha} S)$ je výstup z banky Mel-filtrů warpovaných pomocí VTLN. Pokud definujeme lineární transformaci T^{α}, pak lze napsat

$$L_m^{\alpha} = T^{\alpha} C^{1,00}. \tag{4.25}$$

Vztah mezi $C^{1,00}$ a C^{α} lze z rovnic (4.22) a (4.23) převést do tvaru:

$$C^{\alpha} = (\mathrm{DCT}\ T^{\alpha} \mathrm{DCT}^{-1}) C^{1,00} = W^{\alpha} C^{1,00}, \tag{4.26}$$

kde prvky matice T^{α} jsou podle [65] definovány vztahem

$$T^{\alpha}_{[k,l]} = \frac{1}{2M} \sum_{m=0}^{2M-1} e^{-j\frac{2\pi}{2M}(\frac{\hat{\nu}_m}{\nu_s})k} e^{+j\frac{2\pi}{2M}(\frac{\nu_m}{\nu_s})l} \tag{4.27}$$

s ν_s reprezentující vzorkovací Mel-frekvenci a ν_m (resp. $\hat{\nu}_m$) jsou frekvence (resp. warpované frekvence) jednotlivých Mel-filtrů v bance filtrů a použitím vlastnosti symetrie lze získat $N \times N$ matici $\hat{T}^{\alpha}$ pro následný výpočet transformace W^{α} mezi newarpovaným a warpovaným kepstrem:

$$W^{\alpha} = \mathrm{DCT}\ \hat{T}^{\alpha}\ \mathrm{DCT}^{-1}. \tag{4.28}$$

4.4.2 Odvození VTLN-LT warpováním log-výstupu banky Melovských filtrů

V práci [66] jsou nelineární operátory log a exp z výrazu (4.24) odstraněny použitím aproximace pomocí vhodně zvolené matice pro mapování indexů (viz [67]). Kepstrální tranformace se pak stává lineární:

$$C^{\alpha} = \mathrm{DCT} F_m \alpha F_m^{-1} \mathrm{DCT}^{-1} C^{1,00}). \tag{4.29}$$

Dále je možné přepsat tuto rovnici (4.29) na

$$C^{\alpha} = \mathrm{DCT} L^{\alpha}, \tag{4.30}$$

kde

$$L^{\alpha} = F_m \alpha F_m^{-1} L, \tag{4.31}$$

a

$$L \approx \mathrm{DCT}^{-1} C^{1,00} \tag{4.32}$$

je logaritmus výstupu banky Melovských filtrů.

Pro odvození lineárních transformací založených na warpování spektra podle [66] je s výhodou použitá unitární matice diskrétní kosinové transformace typu II M_{DCT}, která je ortogonální a tudíž pro ni platí $M_{\mathrm{DCT}}^{-1} = M_{\mathrm{DCT}}^{T}$, tedy

$$M_{\mathrm{DCT}} = \left[\beta_k \cos\left(\frac{\pi(2m-1)k}{2M} \right) \right]_{0 \le k \le N-1, 1 \le m \le M}, \tag{4.33}$$

kde M je počet filtrů v bance Melovských filtrů a N je počet kepstrálních koeficientů ve vektoru příznaků a faktor β_k zajišťuje unitárnost matice M_{DCT}

$$\beta_k = \begin{cases} \sqrt{1/M} & \text{pro } k = 0 \\ \sqrt{2/M} & \text{pro } k = 1, 2, \ldots, N-1, \end{cases} \tag{4.34}$$

Poté může být $L = DCT^{-1} C^{1,00} = M_{\mathrm{DCT}}^{T} C^{1,00}$ zapsáno v rozvinuté formě:

$$L(m) = \sum_{k=0}^{N-1} C^{1,00}(k) \beta_k \cos\left(\frac{\pi(2m-1)k}{2M} \right), \; m = 1, 2, \ldots, M \quad , \tag{4.35}$$

S uvažováním kosínové interpolace lze spojité log Mel spektrum $L(u)$, kde u je spojitá proměnná Mel-frekvence, definovat jako

$$L(u) = \sum_{k=0}^{N-1} C^{1,00}(k) \beta_k \cos\left(\frac{\pi(2u-1)k}{2M} \right), \tag{4.36}$$

$$L(m) = L(u)|_{u=m}, \; m = 1, 2, \ldots, M. \tag{4.37}$$

Poté lze aplikovat spojité warpování pomocí warpovací funkce $\psi(u)$ a warpované log Mel-spektrum je

$$L^{\alpha}(m) = L^{\alpha}(u)|_{u=m} = L(\psi(u))|_{u=m} = \sum_{k=0}^{N-1} C^{1,00}(k) \beta_k \cos\left(\frac{\pi(2\psi(m)-1)k}{2M} \right), \; m = 1, 2, \ldots, M \tag{4.38}$$

tedy vektorově zapsáno jako $L^{\alpha} = M_{\mathrm{DCT}}^{\alpha T} C^{1,00}$, kde $M_{\mathrm{DCT}}^{\alpha T}$ je warpovaná matice inverzní diskrétní kosínové transformace a transformované kepstrum lze získat z rovnice:

$$C^{\alpha} = M_{\mathrm{DCT}} \; L^{\alpha} = (M_{\mathrm{DCT}} \; M_{\mathrm{DCT}}^{\alpha T}) C^{1,00} = W^{\alpha} C^{1,00}. \tag{4.39}$$

Výsledná lineární transformace $W^{\alpha} = M_{\mathrm{DCT}} \; M_{\mathrm{DCT}}^{\alpha T}$ je tedy jednodušší pro výpočet.

Vlastní výpočet transformační matice

Spojité log Mel-kepstrum $L(u)$ z rovnice (4.37) je periodické s periodou $2M$ a symetrické okolo bodů $u = 1/2$ a $u = M + 1/2$, z toho důvodu volíme interval u pro warpování jako $1/2 \leq u \leq M + 1/2$. Frekvenční warpovací funkce $\mathcal{F}_\alpha(\omega)$ je ale obvykle definována na intervalu $0 \leq \omega \leq 1$ (více v kapitole 4.3.1), tedy lze najít transformaci mezi u a ω:

$$u \to \omega = \frac{u - 1/2}{M}, \ \frac{1}{2} \leq u \leq M + \frac{1}{2}, \tag{4.40}$$

$$\omega \to u = \frac{1/2}{\omega M}, \ 0 \leq \omega \leq 1. \tag{4.41}$$

Při zohlednění výše zmíněného a s omezením na warpovací funkci ($\tilde{\omega} \in \langle 0, 1\rangle$ a $\mathcal{F}_\alpha(0) = 0$, $\mathcal{F}_\alpha(1) = 1$) lze warpovací funkci $\psi(u)$ z rovnice (4.38) použít ve tvaru

$$\psi(u) = \psi^\alpha(u) = \frac{1}{2} + M\mathcal{F}_\alpha\left(\frac{u - 1/2}{M}\right) \Rightarrow \frac{2\psi(u)^\alpha - 1}{2M} = \mathcal{F}_\alpha\left(\frac{2u - 1}{2M}\right) \tag{4.42}$$

a warpovaná matice DCT z rovnice (4.38) lze přepsat do tvaru

$$M^\alpha_{\mathrm{DCT}} = \left[\beta_k \cos\left(\pi k \mathcal{F}_\alpha\left(\frac{(2m-1)}{2M}\right)\right)\right]_{0\leq k\leq N-1, 1\leq m\leq M}. \tag{4.43}$$

Označíme-li $\omega_m = \frac{2m-1}{2M}$ pro $1 \leq m \leq M$, pak lze rovnice (4.43) a (4.33) přepsat do tvaru:

$$DCT = [[\beta_k \cos(\pi k \omega_m]_{0\leq k\leq N-1, 1\leq m\leq M}, \tag{4.44}$$

$$DCT^\alpha = [\beta_k \cos(\pi k \mathcal{F}_\alpha(\omega_m)]_{0\leq k\leq N-1, 1\leq m\leq M}. \tag{4.45}$$

Poté lze jednoduše vypočítat $W^\alpha = M_{\mathrm{DCT}}\, M^{\alpha T}_{\mathrm{DCT}}$

Vektor pozorování $\boldsymbol{o}$ se obvykle skládá ze statických MFCC koeficientů a prvních a druhých dynamických koeficientů (Δ a $\Delta\Delta$). Lineární transformace dynamických koeficientů je stejná jako statických, transformace celého vektoru pozorování je dána:

$$\boldsymbol{o}^\alpha = A^\alpha \boldsymbol{o} = \begin{bmatrix} C^{1,00} \\ \Delta \\ \Delta\Delta \end{bmatrix} \begin{bmatrix} W^\alpha & 0 & 0 \\ 0 & W^\alpha & 0 \\ 0 & 0 & W^\alpha \end{bmatrix}. \tag{4.46}$$

4.4.3 Odhad optimálního warpovacího faktoru

Parametr α warpovací funkce $\mathcal{F}_\alpha$ je možno odhadovat maximalizací EM optimalizační funkce přes adaptační data [62], tato funkce je shodná s optimalizační funkcí (3.36) vyžívanou se v metodě fMLLR (viz kapitola 3.4.2). Pro akustický model s diagonálními kovariančními maticemi přechází optimalizační funkce na formu (3.37), kde adaptační data jsou nahrazena maticemi akumulovaných statistik (3.38), (3.39) (resp. akumulovanými statistikami definovanými v kapitole 3.2).

Na rozdíl od klasické metody VTLN (viz podkapitola 4.3.2), kdy je pro výpočet kritéria nutné vždy parametrizovat warpovaná adaptační data, stačí v metodě VTLN-LT pouze akumulovat statistiky adaptačních dat, která budou při výpočtu kritéria transformována v závislosti na warpovacím faktoru α.

Prakticky si lze warpovací transformace A^α předpočítat dopředu (závisí pouze na warpovacím parametru α, počtu Mel-filtrů M a počtu kepstrálních koeficientů N, nikoliv však na

konkrétních adaptačních datech) pro vhodnou množinu parametrů α (obvykle v rozmezí $0,88$ až $1,12$ [4]). Optimalizační funkce se pak vyhodnocuje pouze pro předpřipravené transformace $A^{\alpha}|_{0,88 \leq \alpha \leq 1,12}$ a vybere se ta, která maximalizuje pravděpodobnost adaptačních dat (reprezentovaných akumulovanými statistikami daného řečníka). VTLN-LT je proto rychlá metoda vhodná i pro adaptaci s malým množstvím dat, protože odhaduje pouze jeden parametr, α.

Kapitola 5

On-line adaptace

V současné době, kdy jedním z využití systémů ASR je on-line rozpoznávání, nabývá na důležitosti také adaptace za chodu systému (*on-line adaptation*). Tato úloha, na rozdíl od off-line metod adaptace, zahrnuje vyřešení specifických problémů souvisejících s on-line zpracováním mluvené řeči [68]. Při on-line rozpoznávání neznáme dopředu identitu rozpoznávaného řečníka, tedy adaptace musí proběhnout až v průběhu rozpoznávacího procesu na aktuálně rozpoznávaných datech. Hlavním problémem on-line adaptace je obvykle malý tok adaptačních dat kontrastující s požadavkem rychlé adaptace na řečníka. Dále pak absence přepisů (informace od učitele) a možná změna mluvčího v průběhu rozpoznávání.

V této kapitole jsou podrobně popsány tyto problémy a jejich řešení, tedy jmenovitě unsupervised adaptace (v podkapitole 5.1) řešící absenci referenčního přepisu, inkrementální adaptace pro práci s průběžným tokem adaptačních dat (viz podkapitola 5.2) a změna řečníka (podkapitola 5.3).

5.1 Unsupervised adaptace

Adaptační přístupy zmíněné v kapitole 2 využívají obvykle spolu s adaptačními daty i jejich přesný referenční přepis (tzv. informace od učitele). Při on-line adaptaci je však takováto informace nedostupná, a je tedy potřeba adaptovat "bez učitele"(*unsupervised*). Abychom mohli využít dříve zmíněné metody, lze nahradit referenční přepisy adaptačních dat přepisy získanými z prvního průchodu dat rozpoznávačem. Takovýto přepis lze však pouze stěží označit za referenční, protože je zatížen chybou ASR. Aby špatné přepisy neovlivňovaly úspěšnost adaptace, byli navrženy dále popisované metody.

5.1.1 Faktor jistoty (CF)

Přepis adaptačních dat získaných z výstupu rozpoznávače lze jen stěží označit za referenční přepis, protože velmi často obsahuje velké množství chyb. Adaptovat na špatně rozpoznaných datech je kontraproduktivní, proto se společně s přepisem zpracovává i tzv. **faktor jistoty** (CF – Certainty Factor) [69] přiřazený jednotlivým přepsaným slovům. CF je získán s využitím jazykového modelu (LM – Language Model) [19]. Pro adaptaci se využívají jen data, jejichž přepis má vyšší CF než je zvolený práh T_{CF}.

5.1.2 Slovní mřížka

Při uvažování dat s dostatečně velkým CF dochází k redukci adaptačních dat (obvykle jen část dat vyhovuje podmínce dostatečně vysokého CF). Alternativou přepisu s CF pro rozpoznaná slova je využití celé **slovní mřížky** (*lattice*) získané pomocí jazykového modelu. Takovýto přístup upřednostňuje využití všech adaptačních dat před zamítáním nesprávně přepsaných slov. Rozpoznaná data přispívají do statistik (viz kapitola 3.1) hned pro několik HMM stavů s určitou vahou danou slovní mřížkou. Využívá se tedy ne jednoho nejlepšího přepisu, ale hned několika (obvykle N-nejlepších) možných průchodů slovní mřížkou. Tento přístup byl popsán v práci [70] a [69].

5.2 Inkrementální adaptace

Při on-line rozpoznávání máme pro adaptaci k dispozici pouze ta data, která již byla v prvním průchodu rozpoznávačem přepsána. Tato adaptační data přicházejí relativně pomalu a pokud chceme rozpoznávat co nejdříve s adaptovaným modelem, je třeba průběžné adaptace, která by model kontinuálně zlepšovala. Proto jsou v on-line rozpoznávačích využity inkrementální přístupy k adaptaci. Například metodu MAP z podkapitoly 3.3 je možné do inkrementální podoby přetvořit zcela intuitivně. Při MAP adaptaci dochází k posunu (zpřesňování) složek adaptovaného modelu směrem k novým adaptačním datům. Stačí pouze definovat určitý práh dostatečné okupace konkrétní složky modelu a vždy při dosažení tohoto prahu množstvím adaptačních dat složku modelu adaptovat. Zpracovaná data je možno hned zapomenout.

Avšak změna akustického modelu směrem k datům není pro on-line rozpoznávání nejvhodnější, model má obvykle obrovské množství složek ve všech svých stavech, které je časově náročné zpracovávat. Výhodnějším způsobem při on-line adaptaci je změna akustického prostoru rozpoznávaných dat tak, aby lépe odpovídal akustickému modelu. Protože takováto změna je prováděna pomocí transformace dat adaptační maticí, adaptace na řečníka pak znamená pouze změnu adaptační matice nepoměrně menších než je celý akustický model. Hlavním představitelem tohoto přístupu je metoda fMLLR popsaná v kapitole 3.4.2.

5.2.1 Inkrementální fMLLR

V inkrementálním přístupu k metodě fMLLR je potřeba vhodně vyřešit průběžné akumulování statistik (3.2), (3.3) a (3.5) pro jednotlivé složky všech stavů HMM modelu (popř. akumulování jen celkových statistik pro každou z regresních tříd (3.38) a (3.39)). V inkrementálním fMLLR je vhodné pamatovat si všechna data (ve formě statistik), i ta, která již byla využita v předchozím kroku inkrementální adaptace.

Problémem je právě postupné zpřesňování transformace, a tedy průběžná změna akustického prostoru. Dříve nasčítané statistiky jsou v jiném akustickém prostoru než nově nasčítávané, které jsou akumulovány v již transformovaném akustickém prostoru. Řešení této situace je hned několik:

- Nasčítávat k sobě statistiky vypočtené v původním akustickém prostoru s využitím pouze modelu HMM bez transformací. Díky tomu je zachována konzistence statistik v jednotlivých inkrementačních krocích. Ze všech těchto statistik je pokaždé vypočítána nová transformace akustického prostoru. To se používá pouze pro rozpoznávání, nikoliv pro akumulování statistik pro adaptaci.

- K výpočtu nových statistik využít model s transformacemi, ale tyto spočtené statistiky

zpětně transformovat do původního akustického prostoru, abychom je mohli přičíst ke starým statistikám (spočtených právě v původním akustickém prostoru).

- Po spočtení nových transformací (měnících akustický prostor) všechny aktuální statistiky přetransformovat do tohoto akustického prostoru [71]. Pak je možné další statistiky nasčítávat s využitím modelu i transformace a přičítat je ke stávajícím statistikám z předchozí iterace. Transformace statistik do nového akustického prostoru se provádí pomocí matic A a b združených do matice W_{stats}:

$$\boldsymbol{W}_{stats} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ 0 & 1 \end{bmatrix}, \tag{5.1}$$

a transformace statistik je pak dána:

$$\bar{\boldsymbol{\varepsilon}}_{jm}(\boldsymbol{o}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)(\boldsymbol{A}_{(n)}\boldsymbol{o}(t) + \boldsymbol{b}_{(n)})}{\sum_{t=1}^{T} \gamma_{jm}(t)} = \boldsymbol{A}_{(n)}\boldsymbol{\varepsilon}_{jm} + \boldsymbol{b}_{(n)}\,, \tag{5.2}$$

$$\begin{aligned} \bar{\boldsymbol{\varepsilon}}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) &= \frac{\sum_{t=1}^{T} \gamma_{jm}(t)(\boldsymbol{A}_{(n)}\boldsymbol{o}(t) + \boldsymbol{b}_{(n)})(\boldsymbol{A}_{(n)}\boldsymbol{o}(t) + \boldsymbol{b}_{(n)})^{\mathrm{T}}}{\sum_{t=1}^{T} \gamma_{jm}(t)} = \\ &= \boldsymbol{A}_{(n)}\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})\boldsymbol{A}_{(n)}^{\mathrm{T}} + 2\boldsymbol{A}_{(n)}\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o})\boldsymbol{b}_{(n)}^{\mathrm{T}} + \boldsymbol{b}_{(n)}\boldsymbol{b}_{(n)}^{\mathrm{T}}\,. \end{aligned} \tag{5.3}$$

Výpočetně méně náročné je transformovat přímo již celkové akumulované statistiky pro každý shluk regresního stromu (viz podkapitola 3.4.4), kterých je podstatně menší počet než všech složek akustického modelu:

$$\bar{\boldsymbol{k}}_{(n)i} = \boldsymbol{W}_{stats}\boldsymbol{k}_{\boldsymbol{(n)i}}\boldsymbol{W}_{stats}^{T}, \tag{5.4}$$

$$\bar{\boldsymbol{G}}_{\boldsymbol{(n)i}} = \boldsymbol{W}_{stats}\boldsymbol{G}_{\boldsymbol{(n)i}}\boldsymbol{W}_{stats}^{T}, \tag{5.5}$$

kde $\bar{\boldsymbol{G}}_{\boldsymbol{(n)i}}$ a $\bar{\boldsymbol{k}}_{(n)i}$ jsou ztransformované celkové akumulované statistiky i-tého řádku a n-tého shluku a $\boldsymbol{k}_{\boldsymbol{(n)i}}$ a $\boldsymbol{G}_{\boldsymbol{(n)i}}$ jsou aktuálně naakumulované celkové statistiky dané rovnicemi (3.38) a (3.39). $\bar{\boldsymbol{G}}_{\boldsymbol{(n)i}}$ a $\bar{\boldsymbol{k}}_{(n)i}$ jsou pak ekvivalentní statistikám spočteným pomocí modelu s transformacemi. Jedinou aproximací je použití aposteriorních pravděpodobností $\gamma_{jm}(t)$ vypočtených z SI modelu. Ty zůstávají nezměněné (netransformované).

Pamatování si všech transformačních matic spočtených v jednotlivých iteracích (např. ve dvou iteracích matice A_1,b_1 a A_2,b_2) je paměťově nevýhodné, proto se po každé nové iteraci ukládají pouze konečné transformace transformace A_{12},b_{12}:

$$\boldsymbol{A_{12}} = \boldsymbol{A_2}\boldsymbol{A_1}, \tag{5.6}$$

$$\boldsymbol{b_{12}} = \boldsymbol{A_2}\boldsymbol{b_1} + \boldsymbol{b_2}. \tag{5.7}$$

Poslední ze zmiňovaných přístupů k inkrementální adaptaci je pro přesnost výpočtu adaptace nejideálnější (i přes aditivní výpočty), protože každá iterace zpřesňuje akustický prostor jak pro účely rozpoznání, tak i pro přesnější adaptaci.

5.3 Změna řečníka

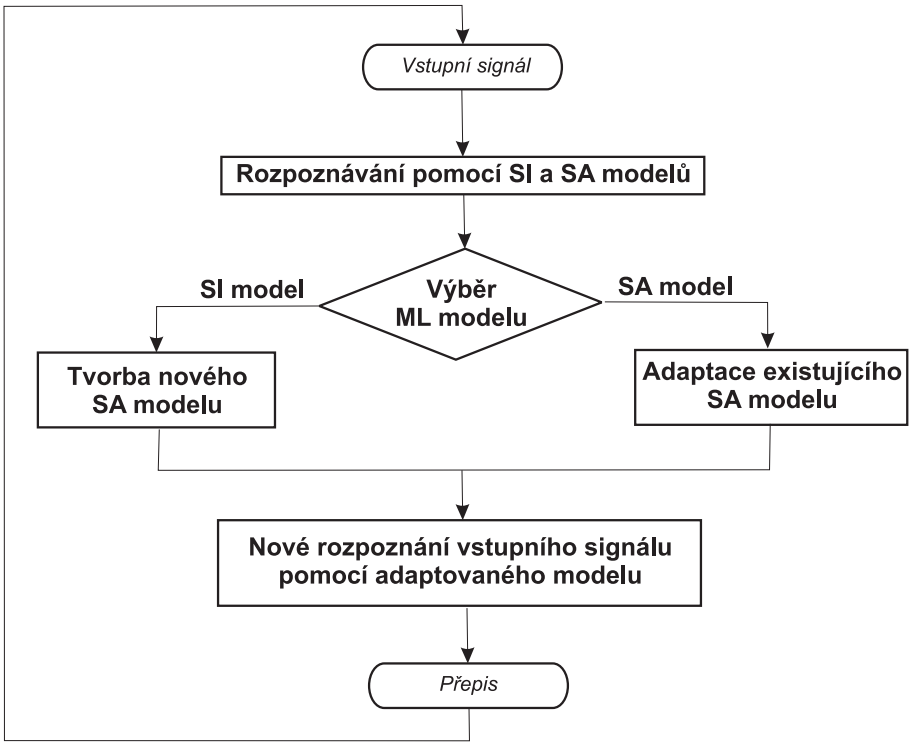
V některých reálných úlohách rozpoznávání řeči může dojít ke změně řečníka v průběhu rozpoznávacího procesu (televizní zprávy, multimediální konference, telefonní hovory, atd.).

Tuto skutečnost lze ignorovat při rozpoznávání s SI modelem. Využíváme-li však adaptaci na konkrétního řečníka, je potřeba odhadnout hranice jeho promluvy a zvolit správný SA model pro rozpoznávání. Tento problém je řešen dvěma nezávislými úlohami: **detekce změny řečníka** (SCD – Speaker Change Detection) [72], [73] a následná **Verifikace/identifikace řečníka** (SV – Speaker Verification) [74].

V on-line systémech ASR je pro detekci změny řečníka největším problémem časová náročnost a nedostatek dat. Z těchto důvodů testujeme změnu v co nejmenších intervalech, ale tak, aby bylo dostatek dat pro identifikaci mluvčího a nalezení jeho změny. Obvykle předem neznáme počet ani identitu řečníků, ani jak často ke změně dochází.

K zjednodušení problému nalezení hranic změny mezi řečníky je možno v některých situacích s výhodou využít **detektor hlasové aktivity** (VAD – Voice Activity Detector) [75], který ze signálu vybere pouze data obsahující řečový signál. Pokud si řečníci neskáčou do řeči, lze tímto způsobem oddělit souvislé segmenty řeči, které lze pokládat za řečené jedním mluvčím. Na to se však nelze v reálných systémech spoléhat, přesto je VAD využíván k odstranění neřečových segmentů signálu.

Jedním z možných řešení adaptace při změně mluvčího je postup navrhovaný v [76], resp. v [77]. Autoři mají k dispozici více akustických modelů (na začátku pouze SI model, resp. více clusterových modelů), přes které provedou rozpoznávání testovaného úseku. Model dávající největší pravděpodobnost je označen za původce promluvy a dále je na tomto úseku adaptován. Pokud největší pravděpodobnost dává jeden z původních modelů, adaptací se založí nový model řečníka. Poté se provede znovu průchod rozpoznávačem s nově adaptovaným model pro získání přesného přepisu rozpoznávaných dat. Tímto postupem, ilustrovaným na obrázku 5.1, lze získat průběžně aktualizované SA modely pro různé řečníky obsažené ve zpracovávaných datech.



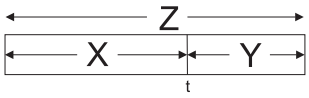
Obrázek 5.1: Struktura systému pro on-line adaptaci při změně řečníka převzatá z práce [76].

Nevýhodou tohoto postupu je jeho časová náročnost. Protože změnu řečníka identifikujeme pomocí maximální věrohodnosti, musíme tedy rozpoznávat testovaná data oproti všem dostupným modelům. Řešením je nahradit tuto část systému jednou z metod detekce změny řečníka.

5.3.1 Detekce změny řečníka (SCD)

V úloze **detekce změny řečníka** (SCD – Speaker Change Detection) jsou často využívány speciální metody parametrizace signálu, které jsou přímo navrženy, aby zdůraznily změnu mluvčího [78]. Pro menší časové zatížení on-line ASR je však vhodné vycházet z již dostupného parametrizovaného signálu (MFCC nebo PLP), aby se snížila náročnost zpracování dat.

V on-line úloze máme k dispozici pouze aktuální a minulá data. Uvažujeme pouze malé množství dat, vybrané okénkem Z o délce N_Z. Naším úkolem je nalézt bod změny t, rozdělující okénko na dvě části X a Y o délce N_X a N_Y (viz obr. 5.2).



Obrázek 5.2: Rozdělení okénka Z na dvě X a Y pro testování změny řečníka t v parametrizovaném signálu.

Úlohu detekce změny řečníka pak formulujeme jako problém testování hypotéz:

H_0: X a Y jsou generovány stejným řečníkem,

H_1: X a Y jsou generovány různým řečníkem. Považujeme-li sekvenci X a Y za náhodný gaussovský proces, lze testování hypotéz převést na úlohu maximalizace věrohodnosti:

L_0: logaritmus pravděpodobnosti, že úsek Z je generován jedním náhodným procesem s parametry θ_Z,

$$L_0 = \sum_{i=1}^{N_X} \log p(x_i|\theta_Z) + \sum_{i=1}^{N_Y} \log p(y_i|\theta_Z), \tag{5.8}$$

L_1: logaritmus pravděpodobnosti, že úseky X a Y jsou generovány dvěma nezávislými náhodnými procesy s parametry θ_X a θ_Y,

$$L_1 = \sum_{i=1}^{N_X} \log p(x_i|\theta_X) + \sum_{i=1}^{N_Y} \log p(y_i|\theta_Y). \tag{5.9}$$

Dvě nejpoužívanější kritéria v úloze detekce změny řečníka jsou **poměr logaritmů pravděpodobnosti** (LLR – Log Likelihood Ratio) [79] a **Bayesovské informační kritérium** (BIC – Bayesian Information Criterion) [80].

Výsledek LLR mezi dvěma okny X a Y je

$$d_{LLR} = L_1 - L_0. \tag{5.10}$$

Logaritmus pravděpodobnosti L_1 bude vždy větší než L_0, je tedy nutné stanovit práh Th, podle kterého rozhodneme, zda došlo ke změně řečníka ($d_{LLR} > Th$).

BIC kritérium se snaží obejít nutnost volby prahu normováním rozdílu logaritmů:

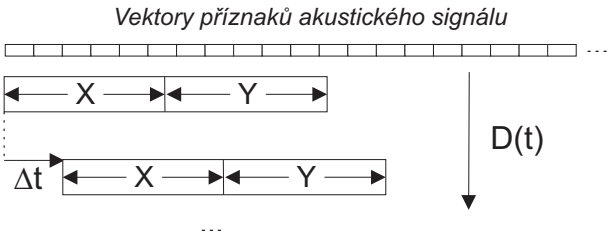
$$d_{BIC} = L_1 - L_0 - \frac{\lambda}{2}\Delta K \log N_Z, \tag{5.11}$$

kde ΔK je rozdíl počtu parametrů modelů a λ je penalizační faktor (obvykle nastavován na 1) [81]. Pokud je $d_{BIC} > 0$, uvažujeme bod t za bod změny.

Postupů, jak procházet signál a nalézt v něm bod změny t, je hned několik [73].

5.3.2 Metoda fixních oken

Široce používaná metoda fixních oken měří statistické vzdálenosti D mezi dvěma sousedními částmi signálu (okénky). Těmito okénky s fixní délkou X a Y se posunujeme po signálu s krokem Δt a zaznamenáváme průběh vzdálenosti v čase (viz obr. 5.3). Za body změny jsou považovány lokální maxima křivky $D(t)$. Často se za statickou vzdálenost D bere Kullback-Leiblerova vzdálenost (např. v [82]), BIC/LLR kritérium je pak použito pro ověření změny.



Obrázek 5.3: Ilustrace metody fixních oken s velikostí oken X a Y s krokem Δt. $D(t)$ označuje vývoj statické vzdálenost v čase.

5.3.3 Metoda binárního dělení

Metoda binárního dělení [73] prochází signál po vzorcích a hledá nejpravděpodobnější změnu řečníka pomocí BIC/LLR kritéria. Pakliže ji najde, rozdělí v jejím místě signál na dva intervaly a rekurzivně pokračuje v hledání změny v těchto intervalech. Algoritmus končí, pokud není v žádném dalším podintervalu nalezena změna.

5.3.4 Metoda s adaptivním oknem

Metoda s adaptivním oknem [80] testuje pomocí BIC/LLR kritéria krátký interval na začátku promluvy, zda neobsahuje bod změny. Pokud tomu tak není, zvětší se prozkoumávaný interval. V opačném případě je začátek nového intervalu označen za nalezený bod změny. Jde o vhodnou metodu pro on-line systémy a lze s ní dosáhnout dobrých výsledků [83].

5.4 Problém malého množství dat

Malé množství adaptačních dat při on-line rozpoznávání způsobuje u adaptačních metod nespolehlivý odhad neznámých parametrů adaptace. Výsledkem je špatně adaptovaný akustický model, který může zhoršit úspěšnost rozpoznávání. Problém robustnosti adaptace s malým množstvím adaptačních dat je podrobně popsán v následující kapitole. Důraz je kladen na adaptační metody založené na lineárních transformacích, které vykazují lepší účinnost v této úloze, protože adaptují i parametry modelu, pro která nebyla v adaptačních datech obsažena žádná pozorování.

Kapitola 6

Robustní adaptace

Předpoklad pro úspěšné použití adaptačních metod je dostatečné množství dat k adaptaci. I metody založené na lineárních transformacích (metody (f)MLLR viz podkapitola 3.4), které byly vyvinuty pro malý počet adaptačních dat, trpí nedostatkem pozorování pro robustní odhad transformačních matic. Při nedostatku dat se stává odhad matic nestabilním a adaptace může zhoršit výsledek rozpoznávání.

Tato kapitola se zabývá různými metodami pro zlepšení odhadu adaptačních parametrů (převážně lineárních transformací) v úloze s malým množstvím dat. Robustní metody jsou založeny například na snížení počtu volných parametrů adaptace, sem patří mimo mj. odhadování pouze diagonálních/blokově diagonálních transformačních matic [23] nebo odhad pouze vektoru posunu a zanedbání rotační matice v lineární transformaci, popsané v podkapitole 6.1. Jiné metody pracují s apriorní informací navíc, např. s dodatečnými statistikami a s vhodnou inicializací odhadu adaptace, viz podkapitola 6.2, nebo s informací z bázového prostoru trénovacích dat uloženou ve vlastních vektorech, viz 6.4, popř. obdobný princip v 6.5.

Tyto postupy jsou mnohdy kombinovány, např. u metod reprezentace transformační matice v prostoru nižší dimenze pomocí bázových vektorů (viz podkapitola 6.6), nižší dimenze redukuje počet odhadovaných parametrů adaptace a samotné bázové vektory slouží jako apriorní informace o prostoru řečníků. V následujících odstavcích je uveden popis známých metod robustní adaptace.

6.1 ShiftMLLR

Původní metoda MLLR popsaná v podkapitole 3.4.1 transformuje střední hodnoty v akustickém modelu dle předpisu

$$\bar{\boldsymbol{\mu}}_{jm} = \boldsymbol{A}_{(n)}\boldsymbol{\mu}_{jm} + \boldsymbol{b}_{(n)}, \tag{6.1}$$

kde $\boldsymbol{\mu}_{jm}$ je původní střední hodnota m-té složky GMM v j-tém stavu HMM, $\bar{\boldsymbol{\mu}}_{jm}$ je adaptovaná střední hodnota, $\boldsymbol{A}_{(n)}$ je transformační matice a $\boldsymbol{b}_{(n)}$ je vektor posunu, vše pro třídu podobných středních hodnot C_n (viz podkapitola 3.4.4), které jsou transformovány stejnou afinní transformací $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$. Pomocí shlukování blízkých středních hodnot dochází k redukování volných parametrů modelu, tedy k odhadování méně parametrů adaptace. Označme dimenzi akustického vektoru d a počet tříd N, pak počet odhadovaných parametrů MLLR adaptace je $N(d^2 + d)$.

Další možností, jak snížit stupeň volnosti adaptace zatímco je zachováno vysoké akustické rozlišení, je použití velkého množství transformací, ale s radikálně nižším počtem odhadovaných parametrů. Pro tento účel byla v [84] popsána metoda **shiftMLLR**, která odhaduje pouze

vektor posunu $\boldsymbol{b}_{(n)}$ středních hodnot, zatímco transformační matici $\boldsymbol{A}_{(n)}$ zanedbává:

$$\bar{\boldsymbol{\mu}}_{jm} = \boldsymbol{\mu}_{jm} + \boldsymbol{b}_{(n)}. \tag{6.2}$$

Tento typ transformace si vystačí s mnohem menším množstvím adaptačních dat, protože počet volných parametrů je výrazně nižší, $d{\cdot}N$. Pro odhad vektoru posunutí $\boldsymbol{b}_{(n)}$ je použito ML-kritérium uvedené v podkapitole 2.3.1, tedy je hledáno maximum optimalizační funkce

$$Q(\lambda, \bar{\lambda}) = -\frac{1}{2} \sum_{b_{jm} \in \lambda} \sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}(t)(\log |\boldsymbol{C}_{jm}| + (\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})^T \boldsymbol{C}_{jm}^{-1}(\boldsymbol{o}^e(t) - \bar{\boldsymbol{\mu}}_{jm})). \tag{6.3}$$

Výsledný hledaný vektor $\boldsymbol{b}_{(n)}$ je dán vztahem:

$$\boldsymbol{b}_{(n)} = (\sum_{e=1}^{E} \sum_{t=1}^{T_e} \sum_{jm \in C_n} \gamma_{jm}(t) \boldsymbol{C}_{jm}^{-1})^{-1} \sum_{e=1}^{E} \sum_{t=1}^{T_e} \sum_{jm \in C_n} \gamma_{jm}(t)[\boldsymbol{C}_{jm}^{-1}(\boldsymbol{o}^e(t) - \boldsymbol{\mu}_{jm})]. \tag{6.4}$$

Pokud je počet regresních tříd C_n volen dynamicky (metoda využívá regresní strom z podkapitoly 3.4.4), lze zvolit mnohem menší práh pro obsazení třídy adaptačními daty, protože odhad vektoru posunutí $\boldsymbol{b}_{(n)}$ je robustnější než odhad afinní transformace $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$ (méně volných parametrů adaptace). V práci [85] pak byl tento přístup rozšířen pro použití v adaptačních technikách pro trénování (tvorba SAT modelu viz podkalitola 4.1).

6.2 Inicializace (f)MLLR

Jednou z možností, jak robustně odhadnout neznámé parametry adaptace při omezeném množství dat, je inicializovat odhad matic nějakou známou hodnotou, například identickou maticí nebo využít metody **zlevněné věrohodné lineární regrese** (DLLR – Discounted Likelihood Linear Regression) [86].

V metodách založených na lineárních transformacích, kde se využívají pro odhad adaptačních transformací $\boldsymbol{W}_{(n)}$ matice akumulovaných statistik $\boldsymbol{k}^s_{(n)}$ a $\boldsymbol{G}^s_{(n)}$ (viz rovnice (3.39) a (3.38)), a kde nedostatek těchto statistik (řídké matice) vede k špatnému odhadu transformací, je vhodné inicializovat matice akumulovaných statistik vhodnou hodnotou za účelem zvýšit robustnost odhadu transformací. V následujících podkapitolách jsou popsány možnosti inicializace různě získanými daty.

6.2.1 Inicializace (f)MLLR statistikami z SI modelu

Efektivní náhradou metody DLLR, která interpoluje adaptační statistiky se statistikami získanými z SI modelu pro vyhnutí se podtečení výsledku odhadu adaptace, je inicializace metod (f)MLLR statistikami SI modelu [71]. Nezačínají se tedy akumulovat statistiky od nuly, ale inicializují se hodnotami získanými z akustického modelu. Dalo by se říci, že matice akumulovaných statistik jsou inicializovány uměle vytvořenými daty, odpovídajícími SI modelu:

$$\boldsymbol{k}_{(n)i} = \sum_{jm \in C_n} \frac{\omega_{jm} \boldsymbol{\mu}_{jm}(i)}{\sigma^2_{jm}(i)} \begin{pmatrix} \boldsymbol{\mu}_{jm} \\ 1 \end{pmatrix}, \tag{6.5}$$

$$\boldsymbol{G}_{(n)i} = \sum_{jm \in C_n} \frac{\omega_{jm}}{\sigma^2_{jm}(i)} \begin{pmatrix} \boldsymbol{\mu}_{jm}\boldsymbol{\mu}_{jm}^T + \boldsymbol{C}_{jm} & \boldsymbol{\mu}_{jm} \\ \boldsymbol{\mu}_{jm}^T & 1 \end{pmatrix}, \tag{6.6}$$

kde ω_{jm}, $\boldsymbol{C}_{jm}$ a $\boldsymbol{\mu}_{jm}$ jsou parametry SI modelu. Při proporcionálním zvětšení všech vah ω_{jm} SI modelu (matice statistik se inicializuje "větším" počtem dat) je (f)MLLR více stabilní, ale méně efektivní. Při výpočtu transformací $\boldsymbol{A}_n, \boldsymbol{b}_n$ pouze z inicializačních dat dostáváme identické matice. Všeobecně lze říci, že při výše zmíněných inicializacích dochází k posílení vlivu původního modelu na úkor informace získané z adaptačních dat.

6.2.2 Využití informace od nejbližších řečníků

Další možností, jak zvýšit množství informace o řečníkovi pro adaptaci, je použít data od akusticky nejvíce podobných osob z trénovací databáze. Jde o rychlou on-line adaptaci, kdy malé množství dat znemožňuje použití klasických adaptačních metod ((f)MLLR, MAP). O této metodě lze uvažovat jako o speciálním případu inicializace, kde na rozdíl od podkapitoly 6.2.1 uvažujeme inicializační data blízká adaptovanému řečníku. Tento postup vychází z principů metody **shlukování mluvčích** (SC – Speaker Clustering) z podkapitoly 3.6 a někdy je též označován jako **kombinace HMM** (*HMM combination*) [87].

V prvním kroku je potřeba vytvořit HMM modely zvlášť pro každého řečníka z trénovací databáze. Nejde v pravém slova smyslu o celý trénovací proces, pouze se provede jedna iterace EM algoritmu původního SI modelu na datech od řečníka z trénovací databáze. Upřesní se pouze statistiky SI modelu - střední hodnota, variance a váha jednotlivých složek. Alternativní možností je využití metod adaptace pro převedení SI modelu na SD model. To je výhodné, pokud je pro daného řečníka málo dat v trénovací databázi. V literatuře je tento krok označován jako získávání HMM statistik.

Společně s HMM modely řečníků se také natrénují jednodušší GMM modely (jednostavové vícesložkové HMM), které při vlastní adaptaci slouží k rychlému nalezení množiny nejpodobnějších řečníků k adaptačním datům. Jednoduchý GMM model slouží dobře i při malém testovacím vzorku, není nutný ani fonetický přepis. Výběr kohorty N nejlepších modelů je prováděn metodami **verifikace řečníka** (SV – Speaker Verification) [88] v závislosti na logaritmu akustické věrohodnosti testovacích dat k jednotlivým GMM. Pro urychlení výběru je někdy vybírána nejprve podmnožina trénovacích řečníků, ze které je dále vybrána finální množina nejbližších (urychlení selekce N-best množiny [89]).

Třetím finálním krokem je vlastní konstrukce adaptovaného HMM modelu, k tomu je využita informace od řečníků z vybrané N-best kohorty.

Kombinace HMM modelů

Nově adaptovaný SD model (viz obr. 6.1) je vypočten z HMM statistik N-best kohorty využitím statistických metod, jde o ekvivalent k jedné iteraci EM trénování SI modelu [90]. Rychlost adaptace souvisí s velikostí N-best kohorty. Při snižování N dochází k značné časové redukci, ale také ke zhoršení kvality adaptovaného modelu pro nedostatečné množství informace. V [91] je tento problém řešen lineární interpolací N-best statistik s globálními statistikami získanými ze všech trénovacích dat:

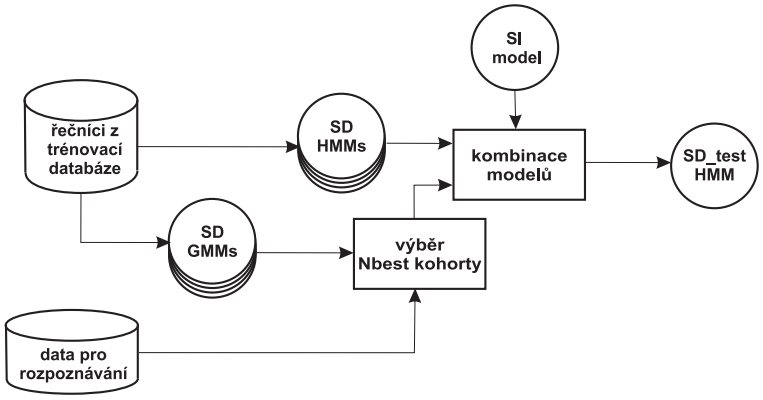
$$\omega_{jm}^{adp_{new}} = \frac{\sum_{n=1}^{N} \gamma_{jm}^{n} + \alpha\gamma_{jm}^{global}}{\sum_{m=1}^{M}\sum_{n=1}^{N} \gamma_{jm}^{n} + \alpha\gamma_{jm}^{global}}, \tag{6.7}$$

$$\boldsymbol{\mu}_{jm}^{adp_{new}} = \frac{\sum_{n=1}^{N} \boldsymbol{\mu}_{jm}^{n} + \alpha\boldsymbol{\mu}_{jm}^{global}}{\sum_{n=1}^{N} \gamma_{jm}^{n} + \alpha\gamma_{jm}^{global}}, \tag{6.8}$$

$$\boldsymbol{\sigma}_{jm}^{adp_{new}} = \frac{\sum_{n=1}^{N} \boldsymbol{\sigma}_{jm}^{n} + \alpha \boldsymbol{\sigma}_{jm}^{global}}{\sum_{m=1}^{M} \sum_{n=1}^{N} \gamma_{jm}^{n} + \alpha \gamma_{jm}^{global}} - \boldsymbol{\mu}_{jm}^{adp_{new}} \boldsymbol{\mu}_{jm}^{adp_{new}T}, \tag{6.9}$$

$$a_{jm}^{adp_{new}} = \frac{\sum_{n=1}^{N} \gamma_{j\to i}^{n} + \alpha \gamma_{j\to i}^{global}}{\sum_{i=1}^{I} \sum_{s=1}^{N} \gamma_{j\to i}^{n} + \alpha \gamma_{j\to i}^{global}}, \tag{6.10}$$

kde ω_{jm}, $\boldsymbol{\mu}_{jm}$, $\boldsymbol{\Sigma}_{jm}$ a a_{ji} jsou parametry HMM modelu (váha, střední hodnota, kovarianční matice m-té složky j-tého stavu a pravděpodobnost přechodu z j-tého stavu do i-tého stavu), kde horní index adp_{new} označuje výsledný adaptovaný model, n označuje statistiky n-tého řečníka z N-best množiny a *global* globální statistiky všech řečníků dohromady. α je empiricky daný váhový faktor.



Obrázek 6.1: Blokové schéma pro adaptaci pomocí kombinace N-best vybraných modelů.

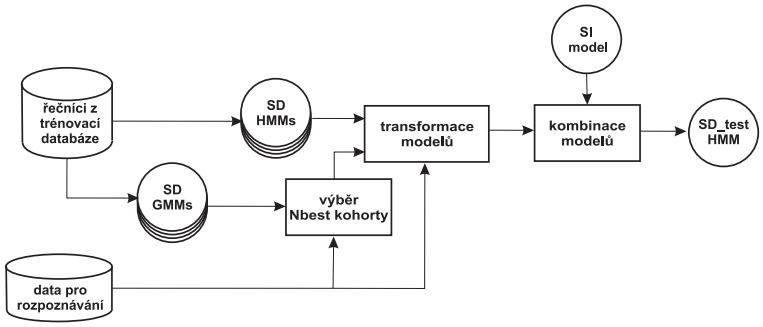
Jiný postup kombinace modelů lze nalézt v [92], kde je uvedena pouze vážená kombinace středních hodnot vybraných HMM modelů. Střední hodnoty výsledného adaptovaného modelu jsou dané vztahem

$$\mu_{jm}^{adp_{new}} = \boldsymbol{M}_{jm} \boldsymbol{\lambda}, \tag{6.11}$$

kde $\boldsymbol{M}_{jm}$ je matice složená ze středních hodnot m-té složky j-tého stavu N-best řečníků. $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_N]^T$ je vektor vah určený z apriorní informace:

$$\lambda_n = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_{jm}^{n}(t)}{\sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_{jm}^{n}(t)}. \tag{6.12}$$

Tento článek také přidává aditivní transformaci N nejlepších modelů před jejich finální kombinací. Autoři tvrdí, že pokud jsou řečníci z trénovací množiny "akusticky daleko"od rozpoznávaného řečníka, charakteristiky vybraných N-best modelů nebudou dostatečně sedět pro tohoto řečníka. Přímá kombinace takovýchto modelů nemusí vyústit v optimální SD model. Proto jsou před vlastní kombinací vybrané modely transformovány (pomocí MLLR) směrem k rozpoznávaným datům, viz obrázek 6.2. Výsledný SD model, vzniklý kombinací takovýchto kompaktnějších modelů, bude více sedět na řečníka, jehož řeč je rozpoznávána. Tento přístup předpokládá dostatečné množství adaptačních dat na robustní odhad MLLR transformace.



Obrázek 6.2: Blokové schéma pro adaptaci pomocí kombinace N-best vybraných modelů s jejich adaptací/transformací směrem k rozpoznávaným datům.

6.3 Apriorní informace z jiné adaptační metody

Namísto inicializace transformačních matic nějakou obecnou hodnotou (v podkapitole 6.2.1) nebo hodnotou blízkou k adaptačním datům (v podkapitole 6.2.2), kde využití této informace znamená utlumení vlivu adaptačních dat, lze apriorní informaci o rozložení adaptačních dat získat z některé méně náročné adaptace [93], jako je např. VTLN-LT (viz podkapitola 4.4) nebo **prediktivní CMLLR** (PCMLLR – Predictive CMLLR) [94], které odhadují jen nepatrné množství volných parametrů (na rozdíl od plné transformační matice).

Tato informace je pak využita pro inicializaci (f)MLLR transformačních matic, tedy v rovnici (3.37) je za transformační matici $\boldsymbol{W}$ dosazena matice odvozená jednoduší metodou (např. VTLN-LT).

Jiná možnost je provést interpolaci nově naakumulovaných statistik $\boldsymbol{k}_{aku}$ a $\boldsymbol{G}_{aku}$ s apriorně vypočítanými $\boldsymbol{k}_{apr}$ a $\boldsymbol{G}_{apr}$ (např. pomocí PCMLLR):

$$\boldsymbol{k}_{(n)i} = \boldsymbol{k}_{aku(n)i} + \tau \frac{\boldsymbol{k}_{apr(n)i}}{\sum_{jm \in C_n} \gamma_{jm}}, \quad \boldsymbol{G}_{(n)i} = \boldsymbol{G}_{aku(n)i} + \tau \frac{\boldsymbol{G}_{apr(n)i}}{\sum_{jm \in C_n} \gamma_{jm}}. \tag{6.13}$$

Využití apriorní informace při výpočtu parametrů transformačních matic zvyšuje robustnost odhadu pro nízké množství adaptačních dat a přitom neomezuje odhad získaný z těchto dat. I jiné metody využívají apriorní informaci, např. shlukování mluvčích z podkapitoly 3.6 nebo metoda MALPR z podkapitoly 3.5.5.

6.4 Vlastní hlasy (EV)

Protože SI akustický model reprezentuje řeč pro univerzálního řečníka, lze intuitivně předpokládat, že model konkrétního řečníka lze reprezentovat v akustickém prostoru menší dimenze. Úkolem je najít systematičtější reprezentaci řeči (charakteristiky příslušné pouze řečníku), která povede k snížení parametrů pro odvození adaptace, a bude tedy robustnější při malém počtu adaptačních dat při zachování variace mezi řečníky.

Jednou z možností, jak najít takové vhodné charakteristiky, je použít tzv. **vlastních hlasů** (EV – Eigen Voices) [95], které byly prvotně použity (pod zkratkou EF – Eigen Face) pro modelování lidské tváře [96]. Vlastní hlasy formují bázi podprostoru, **tzv. prostor vlastních hlasů** (eigenspace), v prostoru parametrů akustického modelu s ohledem na variabilitu mezi řečníky. Myšlenkou je odvodit z množiny trénovacích řečníků malé množství těchto vektorů, které budou reprezentovat různé akustické vlastnosti řečníků (v závislosti na věku, pohlaví,

akcentu, atd.). Model hledaného neznámého řečníka, bod v prostoru vlastních hlasů, bude v tomto podprostoru reprezentován jako lineární kombinace vlastních hlasů.

Chceme-li adaptovat určitou množinu parametrů v modelu (např. střední hodnoty), pak lze tyto parametry zformovat do tzv. supervektoru dimenze D. Z T supervektorů od jednotlivých řečníků z trénovací databáze lze odvodit bázi prostoru dimenze K, tedy množinu vlastních hlasů, $\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{K-1}$, kde $K < T << D$. Tím je omezen prostor, ve kterém je hledán adaptovaný model řečníka.

Adaptovaný supervektor (složený ze středních hodnot $\bar{\boldsymbol{\mu}}_{jm}$ j-tého stavu m-té složky modelu) je počítán jako lineární kombinace vlastních hlasů $\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{K-1}$

$$\bar{\boldsymbol{\mu}} = [\bar{\boldsymbol{\mu}}_1, \ldots, \bar{\boldsymbol{\mu}}_{jm}, \ldots, \bar{\boldsymbol{\mu}}_{JM}]^T = \sum_{i=0}^{K-1} w_i \boldsymbol{e}_i, \tag{6.14}$$

kde $w_0, \ldots, w_{K-1}$ reprezentuje váhy lineární kombinace. Tyto váhy jsou pak adaptačními parametry, které se snažíme pro každého řečníka nalézt.

Před odvozením adaptace je nutné definovat podprostor, ve kterém bude adaptovaný model hledán. Lze natrénovat supervektor středních hodnot akustického modelu pro každého řečníka v trénovací databázi. Ze získaných T supervektorů od jednotlivých řečníků z trénovací množiny odvodíme bázové vektory, které budou reprezentovat podprostor vlastních hlasů. Ideální metoda k tomu určená se nazývá **analýza hlavních komponent**.

6.4.1 Analýza hlavních komponent (PCA)

Analýza hlavních komponent (PCA – Principal Component Analysis) [97] je matematický algoritmus k získání ortogonální transformace, která pokrývá množinu pozorování $\boldsymbol{O}$ (matice, jejíž rozměry jsou $T \times D$ a jejíž proměnné jsou mezi sebou korelovány pro nás neznámým způsobem) a převádí jí na množinu nekorelovaných proměnných, nazývaných hlavní komponenty. Počet hlavních komponent (odpovídá dimenzi podprostoru K) je vždy menší nebo roven počtu původních proměnných, přesněji $K < T << D$.

Transformace je definována tak, aby první hlavní komponenta měla největší rozptyl. Následující hlavní komponenta má pak největší rozptyl za podmínky ortogonality (nekorelovanosti) s komponentou předcházející.

Postup PCA je následující: Trénovací data je nejprve potřeba znormalizovat odečtením jejich střední hodnoty. Poté se spočte z těchto dat kovarianční matice $\boldsymbol{C} = \boldsymbol{O}^T\boldsymbol{O}$ o rozměrech $D \times D$. Provede se rozklad

$$\boldsymbol{V}^{-1}\boldsymbol{C}\boldsymbol{V} = \boldsymbol{\Lambda} \tag{6.15}$$

na matici vlastních čísel $\boldsymbol{\Lambda}$ a vlastní vektory $V = \boldsymbol{e}_0, \ldots, \boldsymbol{e}_{D-1}$, ze který je vybráno K největších (ve smyslu jejich vlastních čísel) hlavních komponent, "vlastních hlasů" $\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{K-1}$. Předpokladem je čtvercová matice $\boldsymbol{C}$, což je kovarianční maticí splněno. Intuitivně vysvětleno, PCA rotuje s původním prostorem tak, aby vlastní vektory byly přidruženy ke směrům s největší variabilitou. Proto je největší variabilita v datech reprezentována K největšími hlavními komponentami, tedy vlastními vektory, se kterými korespondují největší vlastní čísla kovarianční matice pozorování.

6.4.2 Singulární rozklad (SVD)

Dimenze původního prostoru D (odpovídající počtu parametrů modelu) je obvykle obrovská, proto výpočet kovarianční matice trénovacích dat pro PCA (o rozměrech $D \times D$) je velmi ná-

ročný. Problém lze zjednodušit použitím metody, která se nazývá **singulární rozklad** (SVD – Singular Value Decomposition) [98]. Tato metoda spočívá v zobecněném rozkladu libovolné matice $\boldsymbol{O}$ na tzv. singulární vektory $\boldsymbol{U}$ a $\boldsymbol{V}$:

$$\boldsymbol{O} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{6.16}$$

Navíc sloupce matice $\boldsymbol{U}$ $(T \times T)$ a $\boldsymbol{V}$ $(D \times D)$ jsou vlastní vektory matice $\bar{\boldsymbol{C}} = \boldsymbol{O}\boldsymbol{O}^T$, resp. matice $\boldsymbol{C} = \boldsymbol{O}^T\boldsymbol{O}$. Protože vektory matic $\boldsymbol{U}$ a $\boldsymbol{V}$ jsou ortogonální, lze napsat [99]

$$\boldsymbol{O}^T\boldsymbol{O}\boldsymbol{V} = \boldsymbol{O}^T\boldsymbol{U}\bar{\boldsymbol{\Sigma}}, \tag{6.17}$$

kde $\bar{\boldsymbol{\Sigma}}$ má stejné vlastnosti jako $\boldsymbol{\Sigma}$. Lze snadno nahlédnout, že prvních T vektorů matice $\boldsymbol{V}$ lze odvodit z $\boldsymbol{O}^T\boldsymbol{U}$. Tato vlastnost je využita pro výpočet T hlavních komponent $\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{T-1}$, kde $\boldsymbol{e}_j = \boldsymbol{O}^T\boldsymbol{u}_j$ a $\{\boldsymbol{u}_j\}_{j=\ldots T-1}$ jsou vlastní vektory matice $\bar{\boldsymbol{C}} = \boldsymbol{O}\boldsymbol{O}^T$. Výpočetní náročnost matice $\bar{\boldsymbol{C}} = \boldsymbol{O}\boldsymbol{O}^T$ je značně nižší než kovarianční matice $\boldsymbol{C} = \boldsymbol{O}^T\boldsymbol{O}$, za rozumného předpokladu $K < T << D$.

6.4.3 Dekompozice vlastních hlasů (ED)

Proces adaptace využívající vlastní hlasy $E = [\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{T-1}]$ se nazývá **dekompozice vlastních hlasů** (ED – Eigenvoices Decomposition) [100]. Jde o aplikaci EM algoritmu, v literatuře je někdy též označován jako **maximální věrohodnost dekompozice vlastních hlasů** (MLED – Maximal Likelihood Eigenvoices Decomposition). Princip spočívá v nalezení váhového vektoru $\boldsymbol{w} = [w_0, \ldots, w_{K-1}]$ tak, aby $\boldsymbol{E}\boldsymbol{w}$ vytvořilo nový supervektor středních hodnot akustického modelu, který bude maximalizovat věrohodnost adaptačních dat. Odvození vyplývá ze standardního trénování akustického modelu EM algoritmu aplikací Baum-Welchova reestimačního algoritmu (viz podkapitola 2.3.1). Maximalizuje se pomocná funkce (2.21) s omezením na akustický model λ ležící v podprostoru vlastních vektorů (jeho střední hodnoty jsou dány lineární kombinací vlastních vektorů). Pomocnou funkci lze převést na tvar

$$Q(\lambda, \bar{\lambda}) = -\frac{1}{2}P(\boldsymbol{O}|\lambda)\sum_{j}^{J}\sum_{m}^{M}\sum_{t}^{T}\gamma_{jm}(t)(nlog(2\pi) + log|\boldsymbol{C}_{jm}| + (o(t) - \bar{\boldsymbol{\mu}}_{jm})^T\boldsymbol{C}_{jm}^{-1}(o(t) - \bar{\boldsymbol{\mu}}_{jm})). \tag{6.18}$$

Množinu vah $w_0, \ldots, w_{K-1}$ odvodíme maximalizací této pomocné funkce, $\partial Q/\partial w_i = 0$ pro $i = 0, \ldots, K-1$.

V článku [101] je k odvození vlastních hlasů využita nelineární metoda PCA, ale pomocí kernel transformace přechází problém na lineární PCA. Tato metoda je nazývána **dekompozice vlastních hlasů využitím kernelu** (KEV – Kernel Eigen Voices).

6.4.4 EigenMAP

Aplikace problému vlastních hlasů do adaptace typu MAP (MAP adaptace je podrobně popsána v podkapitole 3.3) lze nalézt např. v [102]. Autor předpokládá populaci S řečníků a chce pro ně zkonstruovat model (ať již GMM pro SV nebo HMM pro ASR) s využitím apriorní informace z SI modelu, kde $\boldsymbol{\mu}_{jm}$ je střední hodnota j-tého stavu a m-té složky na řečníku nezávislém SI akustickém modelu. Za předpokladu MAP přístupu je pro konkrétního řečníka s adaptovaná složka modelu $\boldsymbol{\mu}_{jm}^s$ dána

$$\boldsymbol{\mu}_{jm}^s = \boldsymbol{\mu}_{jm} + \boldsymbol{O}_{jm}^s, \tag{6.19}$$

kde $\boldsymbol{O}_{jm}^s$ je nepozorovatelný vektor posuvu se známou apriorní distribuční maticí $(\hat{\boldsymbol{O}}_{jm}^s)$, pro kterou platí:

- klasické MAP předpokládá, že položky matice $\hat{\boldsymbol{O}}^s_{jm}$ jsou statisticky nezávislé,
- eigenMAP předpokládá vektory matice $\hat{\boldsymbol{O}}^s_{jm}$ **nezávislé a identicky distribuované** (iid – independent and identically distributed).

V případě eigenMAP lze pak najít podprostor původního prostoru této matice, ve kterém je odhad $\boldsymbol{O}^s_{jm}$ na adaptačních datech méně náročný. Podprostor je dán nejinformativnějšími vlastními vektory původního prostoru. Postup je popsán v [103].

6.4.5 EigenMLLR

Stejně jako MAP, tak i adaptační metody založené na lineárních transformacích (viz podkapitola 3.4) lze upravit pro využití metody vlastních hlasů. Jelikož počet volných parametrů metod LT závisí na množství adaptačních dat, jsou tyto metody vhodnější pro robustní adaptaci, a tedy i pro kombinaci s přístupem využívajícím vlastní hlasy. Mezi LT přístupy patří především (f)MLLR adaptace shlukující podobné parametry adaptovaného modelu, ty jsou pak transformovány stejnou adaptační maticí.

V článku [99] je popsána metoda **eigenMLLR**, která k odvození transformační matice $\boldsymbol{W}$ pro adaptovaného řečníka používá apriorní informaci reprezentovanou vlastními hlasy odvozenými z transformačních matic S řečníků z trénovací databáze. Tyto trénovací matice $\boldsymbol{W}^s$ jsou nejprve pospojovány po řádcích do supervektorů $\mathrm{vec}(\boldsymbol{W}^s)$ a poskládány do matice $\boldsymbol{Z}$. Z této matice supervektorů $\boldsymbol{Z}$ je pak PCA přístupem (viz podkapitola 6.4.1) odvozeno K největších vlastních vektorů, vlastních hlasů $\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{K-1}$. Výsledná adaptační matice $\bar{\boldsymbol{W}}$ pro adaptovaného řečníka je dána lineární kombinací vlastních hlasů

$$\mathrm{vec}(\bar{\boldsymbol{W}}) = \sum_{i=0}^{K-1} w_i \boldsymbol{e}_i. \tag{6.20}$$

K odhadu váhových koeficientů $w_0, \ldots, w_{K-1}$ lze použít MLED kritérium, pokud nejprve adaptujeme střední hodnoty SI modelu pomocí odvozené transformační matice $\bar{\boldsymbol{W}}$ a tyto adaptované střední hodnoty spojíme do supervektoru. Pak je problém řešen stejně jako v podkapitole 6.4.3. Zobecněný přístup využívající lineární kombinaci je podrobně popsán v podkapitole 6.6.

Počet transformačních matic pro adaptovaného řečníka lze volit v závislosti na počtu dat (viz podkapitola 3.4.4). Pro eigenMLLR můžeme volit mnohem menší okupační práh pro jednotlivé třídy (stačí méně adaptačních dat) než v klasickém MLLR, protože pracují s apriorní informací danou EV. Rozšíření tohoto přístupu o nelineární PCA nebo eigenMAPLR přístup lze nalézt v článcích [104] resp. [105].

6.5 Faktorová analýza (FA)

Další z metod, která hledá podprostor původního akustického prostoru pro zdůraznění variance mezi řečníky, je **faktorová analýza** (FA – Factor Analysis) [106]. Jde o statistickou metodu, která popisuje variability mezi pozorovanými korelovanými proměnnými menším množstvím nepozorovaných, nekorelovaných proměnných, zvaných faktory (latentní proměnné). FA hledá spojení mezi proměnnými pro zjištění nepozorovaných latentních proměnných. Pozorované proměnné jsou modelované jako lineární kombinace potenciálních faktorů (s předpokladem určité chyby). Informace o závislostech pozorovaných proměnných získaná pomocí FA lze poté použít k redukování množství proměnných v pozorovaných datech. FA pracuje na obdobném

principu jako PCA, ale s tím rozdílem, že FA testuje hypotézu za předpokladu chyby, zatímco EV dekompozice pomocí PCA je popisnou statistickou metodou [107].

FA je založena na korelačních a parciálních korelačních koeficientech. Korelační koeficient vyjadřuje těsnost - lineární závislosti proměnných. Je-li možné závislost dvou proměnných vysvětlit společným faktorem, musí být parciální korelační koeficient, kde je tento vliv ostatních faktorů odrušen, blízký nule.

Je-li dána množina pozorovaných náhodných proměnných $o_1, \ldots, o_D$ se střední hodnotou $\mu_1, \ldots, \mu_D$, pak lze vyslovit hypotézu, že existuje neznámá konstanta l_{ij} a K nepozorovaný počet náhodných proměnných F_j (tzv. faktorů), kde $i \in 1, \ldots, D$, $j \in 1, \ldots, K$ a $K < D$. Lze tedy napsat:

$$o_i - \mu_i = l_{i1}F_1 + \cdots + l_{ik}F_K + \varepsilon_i, \tag{6.21}$$

v maticovém zápisu pro T pozorování

$$\boldsymbol{O} - \boldsymbol{\mu} = \boldsymbol{L}\boldsymbol{F} + \boldsymbol{\varepsilon}, \tag{6.22}$$

kde $\boldsymbol{\varepsilon}$ je nezávisle distribuovaná chyba, část původní proměnné, o níž předpokládáme, že její korelace se všemi faktory je nulová, má tedy normální rozdělení

$$\mathrm{Cov}(\boldsymbol{\varepsilon}) = \mathrm{Diag}(\psi_1, \ldots, \psi_K) = \boldsymbol{\Psi} \text{ and } \mathrm{E}(\boldsymbol{\varepsilon} = 0), \tag{6.23}$$

dále pak je $\boldsymbol{O}_{D\times T}$ matice pozorování, $\boldsymbol{L}_{D\times K}$ matice faktorových zátěží a $\boldsymbol{F}_{K\times T}$ matice faktorů. Pro $\boldsymbol{F}$ platí předpoklady:

- $\boldsymbol{F}$ a $\boldsymbol{\varepsilon}$ jsou vzájemně nezávislé,
- $\boldsymbol{F}$ má normální rozdělení $\mathcal{N}(\boldsymbol{F}) = (0, I)$.

Faktory jsou tedy konstruovány tak, aby spolu vzájemně nekorelovaly.

Označme $\mathrm{Cov}(\boldsymbol{O} - \boldsymbol{\mu}) = \Sigma$, pak lze z výše uvedených předpokladů odvodit

$$\Sigma = \boldsymbol{L}\mathrm{Cov}(\boldsymbol{F})\boldsymbol{L}^T + \mathrm{Cov}(\boldsymbol{\varepsilon}) \tag{6.24}$$

a

$$\Sigma = \boldsymbol{L}\boldsymbol{L}^T + \Psi. \tag{6.25}$$

Faktorová analýza se realizuje pouze na výběru ze zkoumaného prostoru, proto budou analogicky výsledky faktorové analýzy pouze odhady skutečných faktorů. Pro extrakce faktorů existuje několik metod, spočívají v určení počtu faktorů a velikostí faktorových zátěží. Mezi tyto metody patří např. **metoda hlavních os**, **metoda nejmenších čtverců**, **metoda hlavních komponent**.

6.5.1 Spojená faktorová analýza (JFA)

Rozšíření metody FA o další nezávislé faktory se nazývá **spojená faktorová analýza** (JFA – Joint Factor Analysis) [108], [109] původně vyvinuta pro úlohu rozpoznávání řečníka. Předpokladem jsou dvě různé variability v datech, tedy např. variabilita jak v řečníku, tak i v kanálu . Metoda JFA předpokládá rozklad

$$\boldsymbol{M} = \boldsymbol{s} + \boldsymbol{c} = \boldsymbol{s} + \boldsymbol{u}\boldsymbol{x}, \tag{6.26}$$

kde $\boldsymbol{M}$ je supervektor odpovídající aktuálním datům, $\boldsymbol{s}$ je na řečníku závislý supervektor a $\boldsymbol{c}$ závisí pouze na vlastnostech kanálu, $\boldsymbol{u}$ je čtvercová matice faktorových zátěží a $\boldsymbol{x}$ jsou faktory kanálu s normálním rozdělení. V případě JFA je supervektor $\boldsymbol{s}$ modelován jako

$$\boldsymbol{s} = \boldsymbol{m} + \boldsymbol{v}\boldsymbol{y} + \boldsymbol{d}\boldsymbol{z}, \tag{6.27}$$

kde $\boldsymbol{m}$ je supervektor nezávislý ani na řečníku ani na kanálu, $\boldsymbol{v}$ je čtvercová matice a $\boldsymbol{d}$ je diagonální matice, $\boldsymbol{y}$ a $\boldsymbol{z}$ jsou náhodné vektory s normálním rozdělením (nazývané běžné a specifické faktory). Absence specifických faktorů implikuje, že informace o testovaných supervektorech je obsažena pouze v $\boldsymbol{m}$ a $\boldsymbol{v}$, což je předpoklad metody EV z podkapitoly 6.4.

Z JFA vychází metoda **iVectorů** (*iVectors*) [110], která se také převážně využívá v úloze rozpoznávání řečníka s daty nahranými na různých akustických kanálech. Tento stav obvykle v ASR nenastává, rozpoznáváme jednoho řečníka na stejném kanálu (kanál a řečník splývají do sebe), proto jsou v adaptaci obvykle používané pouze metody EV nebo FA.

6.6 Reprezentace transformace v prostoru nižší dimenze pomocí bázových vektorů

Další možný postup, jak se vypořádat s malým množstvím adaptačních dat při adaptaci (konkrétně při metodách založených na lineárních transformacích, kde počet neznámých parametrů je dán $D = d \times (d + 1)$, d je dimenze vektoru pozorování), je založen na reprezentaci transformační matice v nižším podprostoru definovaném pomocí bázových matic, metoda tzv. **bázové reprezentace** [111]. Hledaná adaptační matice $\boldsymbol{W}$ konkrétního rozpoznávaného řečníka je dána lineární kombinací bázových vektorů:

$$\boldsymbol{W} = \boldsymbol{W}_0 + \sum_{b=1}^{B} \alpha_b \boldsymbol{W}_b, \tag{6.28}$$

kde $\boldsymbol{W}_0 = [\boldsymbol{I}; \boldsymbol{0}]$ a $\boldsymbol{W}_b$ jsou bázové matice, které určují podprostor hledání transformace $\boldsymbol{W}$. Bázové matice $\boldsymbol{W}_b$ jsou apriorní znalostí o daném podprostoru a omezují tím adaptaci na hledání pouze váhových koeficientů α_b. Váhy α_b jsou závislé na aktuálním řečníku. B je dimenze podprostoru v rozsahu $1 \leq B << D$. Bázové matice jsou odvozovány před vlastní adaptací z testovacích dat. Jediné parametry, které hledáme při adaptačním procesu, jsou počet B a velikosti váhových koeficientů α_b, jejichž počet je podstatně nižší, než původní počet neznámých D. Jejich odvození z malého množství dat je robustnější než odhad všech parametrů transformační matice. B je voleno v závislosti na velikosti adaptační množiny.

6.6.1 Volba bázových matic

Bázové matice jsou určovány z trénovacích dat před započtením adaptačního procesu, tedy bez znalosti dat adaptovaného řečníka. Níže jsou popsány postupy odhadu bázových matic.

Báze definovaná dekompozicí vlastních vektorů

Jednou z možností, jak volit bázi pro lineární kombinaci, je v podkapitole 6.4.5 uvedená metoda dekompozice vlastních hlasů / vlastních vektorů [99]. Je nutné najít vlastní vektory kovarianční matice vstupních dat (o velikosti T vzorků). Vstupní data jsou dána supervektory $\boldsymbol{w}^s = \mathrm{vec}(\boldsymbol{W}^s)$ sestavenými z transformačních matic $\boldsymbol{W}^s$ trénovacích řečníků $1...s$, kde operátor vec pospojuje řádky matice do jediného supervektoru. Dimenze supervektoru je $D >> T$.

Za účelem nalézt vlastní vektory definujeme kovarianční matici $\boldsymbol{Z}^T\boldsymbol{Z}$, kde $\boldsymbol{Z}$ je $T \times D$ matice vstupních dat (supervektorů) poskládaných do sloupců a normalizovaných na nulovou střední hodnotu. Pro lineární kombinaci (6.28) je využito pouze B vlastních vektorů s největšími odpovídajícími vlastními čísly.

ML odhad

V práci [112] byl odvozen přístup pro hledání bázových matic vycházející z ML kritéria. Tento přístup vychází z práce [113], kde bylo navrženo odvození bázových matic s využitím EM algoritmu, který však není vhodný pro rychlou adaptaci. Proto je v práci [112] navržen rychlejší přístup, kdy odhad pomocí ML kritéria přechází za určitých předpokladů na rychlejší dekompozici vlastních vektorů.

Při tomto přístupu je třeba opět transformační matici $\boldsymbol{W}$ přeorganizovat do tvaru supervektoru $\boldsymbol{w} = \mathrm{vec}(\boldsymbol{W})$, kde operátor vec poskládá řádky matice $\boldsymbol{w_j}, j = 1, \ldots, J$ do sloupcového supervektoru $\boldsymbol{w} = [\boldsymbol{w_1}^T, \ldots, \boldsymbol{w_J}^T]^T$. Vezmeme-li v úvahu Taylorův rozvoj druhého řádu pomocné funkce (3.37) pro $\boldsymbol{w}$, $\boldsymbol{w} = \boldsymbol{w}_0$, dostáváme

$$\boldsymbol{Q}^s(\boldsymbol{w}) = (\Delta\boldsymbol{w})^{\mathrm{T}}\boldsymbol{p}^s - \frac{1}{2}(\Delta\boldsymbol{w})^{\mathrm{T}}\boldsymbol{H}^s(\Delta\boldsymbol{w}), \tag{6.29}$$

kde $\Delta\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}_0$ a $\boldsymbol{p}^s$, $\boldsymbol{H}^s$ jsou počítány z matric akumulovaných statistik $\boldsymbol{k}_i^s$, $\boldsymbol{G}_i^s$ a β^s pro daného řečníka s. Idea vychází ze změny proměnné, tak aby po přepsání bylo $\boldsymbol{H}^s$ dobře podmíněno (tj. blízké jednotkové matici vynásobené konstantou). Proto je definováno

$$\boldsymbol{H} = \frac{1}{\sum_s \beta^s}\sum_s \boldsymbol{H}^s, \tag{6.30}$$

jako průměrná hodnota $\boldsymbol{H}^s$ normalizovaná počtem pozorování. Poté lze získat z Choleskiho rozkladu dolní trojúhelníkovou matici $\boldsymbol{C}$, $\boldsymbol{H} = \boldsymbol{C}\boldsymbol{C}^T$ a provést změnu proměnné $\hat{\boldsymbol{w}} = \boldsymbol{C}^T\boldsymbol{w}$. Po dosazení do (6.29) dostáváme tvar pomocné funkce

$$\hat{\boldsymbol{Q}}^s(\hat{\boldsymbol{w}}) = (\Delta\hat{\boldsymbol{w}})^{\mathrm{T}}\hat{\boldsymbol{p}}^s - \frac{1}{2}(\Delta\hat{\boldsymbol{w}})^{\mathrm{T}}\hat{\boldsymbol{H}}^s(\Delta\hat{\boldsymbol{w}}), \tag{6.31}$$

kde $\Delta\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_0$, $\hat{\boldsymbol{w}}_0 = \boldsymbol{C}^T\boldsymbol{w}_0$ a dále

$$\hat{\boldsymbol{H}}^s = \boldsymbol{C}^{-1}\boldsymbol{H}^s\boldsymbol{C}^{-T} \tag{6.32}$$

a

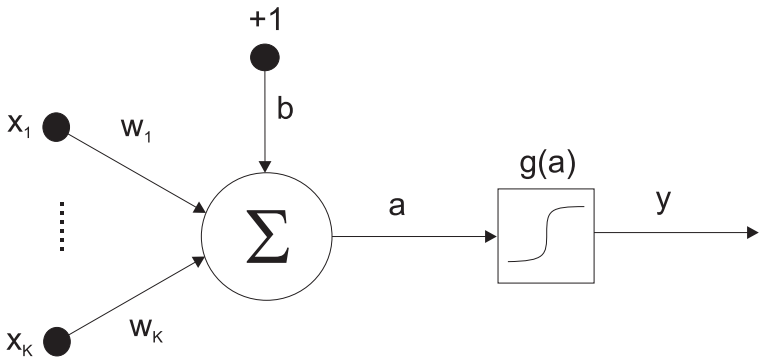
$$\hat{\boldsymbol{p}}^s = \boldsymbol{C}^{-1}\boldsymbol{p}^s. \tag{6.33}$$

Pak lze říci, že $\hat{\boldsymbol{H}} = \boldsymbol{I}$, což byla motivace pro změnu proměnné.

Při uvažování, že všichni řečníci jsou si dostatečně podobní, přidáme zjednodušení $\boldsymbol{H}^s \cong \beta^s\boldsymbol{H}$ a ekvivalentně $\hat{\boldsymbol{H}}^s \cong \beta^s\boldsymbol{I}$, což je nutné pro redukci ML problému na problém PCA, který je lépe časově zvládnutelný.

Lze dokázat, že při omezení $\boldsymbol{w}$ na formu lineárních kombinací bází (6.28), je funkce (6.31) maximální při uvažování bázových vektorů $\boldsymbol{w}_b = \mathrm{vec}(\boldsymbol{W}_b)$ spočítaných pomocí dekompozice vlastních vektorů z matice $\boldsymbol{M}$, která je dána vztahem

$$\boldsymbol{M} = \sum_s \frac{1}{\beta^s}\boldsymbol{p}^s\boldsymbol{p}^{sT}. \tag{6.34}$$



Obrázek 6.3: Model neuronu, tzv. perceptron s K vstupy x_k a s aktivační/přenosovou funkcí $y = g(a)$.

6.6.2 Hledání váhových koeficientů

Váhové koeficienty α_b jsou závislé na adaptovaném řečníkovi, je tedy nutné je najít při adaptaci. V článku [112] je popsán postup hledání váhových koeficientů pomocí maximalizace pomocné funkce (3.37) metodou gradientního poklesu [114].

6.7 Redukce informace pomocí neuronové sítě

6.7.1 Neuronová síť (ANN)

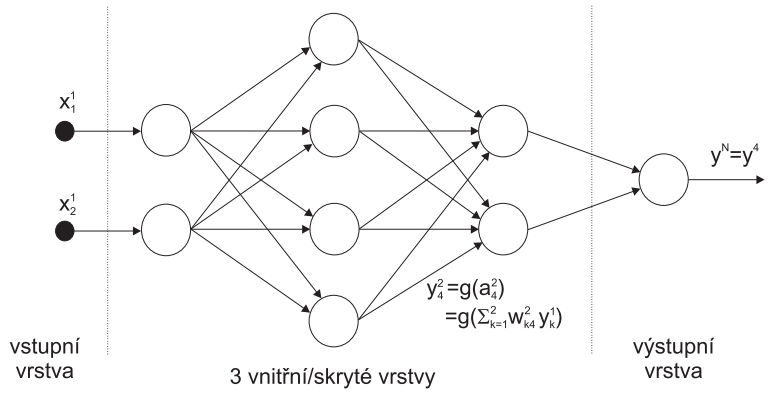
Umělé neuronové síťe (ANN – Artifcial Neural Networks) [115] mají vzor v chování odpovídajících biologických struktur. Využívají se při zpracování informace vyznačující se distribuovaným paralelním zpracováním dat. Struktura umělé neuronové sítě je složena z umělých neuronů simulujících funkci biologického neuronu. Tyto neurony si v ANN navzájem předávají informaci, kterou transformují pomocí v sobě implementovaných přenosových funkcí. Model neutronu, který má vždy několik vstupů, ale pouze jeden výstup, je naznačen na obrázku 6.3. Jde o tzv. perceptron [116].

Funkce neuronu je následná: neuron obdrží stimul od některého okolního neuronu připojeného k některému jeho vstupu. Více stimulů je neuronem zkombinováno dohromady, s respektováním důležitosti (váhy) jeho vstupů. Když kombinace vstupních signálů dosáhne určité hodnotu, neuron je aktivován a přes jeho výstup je vyslán stimul k následným neuronům v síti. Funkci perceptronu lze tedy popsat rovnicí

$$y = g(a) = g(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b) = g(\sum_{k=1}^{K} w_k x_k + b), \tag{6.35}$$

kde y je výstupem perceptronu, $\boldsymbol{x} = [x_1, \ldots, x_K]^T$ je vstupní vektor, $\boldsymbol{w} = [w_1, \ldots, w_K]^T$ je váhový vektor jeho vstupů a b je aktivační práh (aditivní vektor). Aktivační/přenosová funkce g je obecně nelineární funkcí.

Neurony jsou spojovány do sítí přes své vstupy a výstupy. Jedním typem takové sítě je i vrstvená perceptronová neuronová síť (MLP-ANN – Multi-layer Perceptron ANN) [117], kde jsou neurony sdruženy do tzv. vrstev (vstupní, výstupní a více vnitřních/skrytých), viz příklad na obrázku 6.4. N-vrstvá síť MLP-ANN pracuje diskrétně, signál je propagován z jedné vrstvy n striktně pouze do vrstvy následující $n + 1$. Výstup všech K_n perceptronů y_k^n z n-té vrstvy



Obrázek 6.4: Umělá neuronová síť se 4 vrstvami.

lze poskládat do výstupního vektoru dané vrstvy $y^n = [y_1^n, \ldots, y_{K_n}^n]$, kde y_k^n je výstup k-tého neuronu v n-té vrstvě. Platí, že výstup n-té vrstvy je vstupem vrstvy $n+1$, tedy $y^n \equiv x^{n+1}$. Vstup sítě MLP-ANN $y^0 \equiv x^1$ je reprezentován hypotetickou vstupní vrstvou 0 a výstup y^N pak výstupní vrstvou N. Ostatní vrstvy $1 \ldots (N-1)$ se pak nazývají vnitřní nebo také skryté vrstvy.

Pro výstup n-té vrstvy platí

$$y^n = g(\boldsymbol{a}^n) = g(\boldsymbol{y}^{(n-1)T}\boldsymbol{W}^n + \boldsymbol{b}^n), \tag{6.36}$$

kde $\boldsymbol{W}^n$ je váhová matice n-té vrstvy sítě o rozměrech $K_n \times K_n$, jejíž k-tý sloupec je tvořen váhami k-tého neuronu v n-té vrstvě. Aditivní vektor je dán jako $\boldsymbol{b}^n = [b_1^n, \ldots, b_{K_n}^n]$ a $\boldsymbol{a}^n = [a_1^n, \ldots, a_{K_n}^n]$.

Aktivační funkce

Aktivační funkce g_k^n k-tého neuronu v n-té vrstvě je obvykle nelineární funkcí, která může být obecně různá pro různé neurony v různých vrstvách, obvykle se však používá stejná aktivační funkce u všech neuronů v dané vrstvě. Možné aktivační funkce mohou být například:

- **Skoková funkce**

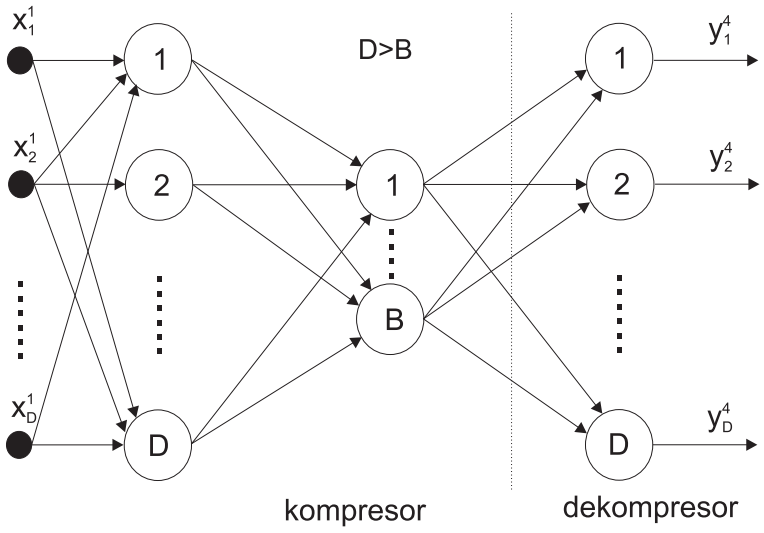
$$g_k^n(\boldsymbol{a}^n) = \begin{cases} 0 \text{ pro } a_k^n < 0 \\ 1 \text{ pro } a_k^n \geq 0 \end{cases}, \tag{6.37}$$

- **Sigmoidální funkce**

$$g_k^n(\boldsymbol{a}^n) = \frac{1}{1 + \exp(-a_k^n)}. \tag{6.38}$$

Trénování ANN

Ačkoliv má neuronová síť mnoho volných parametrů, obvykle je její topologie daná apriori. Trénováním je hledáno pouze nejvhodnější nastavení vah jednotlivých neuronů $\boldsymbol{W} = [\boldsymbol{W}^1, \ldots, \boldsymbol{W}^N, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_N]$. Trénovací proces může být založen na informaci od učitele (supervised) nebo bez ní (unsupervised). Pro rozsah této práce přichází v úvahu pouze trénování s učitelem. To znamená, že při trénovacím procesu máme k dispozici trénovací páry



Obrázek 6.5: Topologie umělé neuronové sítě bottleneck.

$\aleph = [\boldsymbol{x}_t, \boldsymbol{y}^{*N}(\boldsymbol{x}_t)]_{t=1}^{T}$, tedy vstup s jeho žádaným výstupem poslední vrstvy. Úkolem je najít váhy sítě $\boldsymbol{W}$, které minimalizují ztrátovou funkci $E(\aleph|\boldsymbol{W})$, kde ztrátová funkce je dána např. jako minimální kvadrát chyby

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} E(\aleph|\boldsymbol{W}) = \sum_{t=1}^{T} E_t = \sum_{t=1}^{T} 1/2 \left\| \boldsymbol{y}^N(\boldsymbol{x}_t) - \boldsymbol{y}^{*N}(\boldsymbol{x}_t) \right\|, \tag{6.39}$$

kde $\boldsymbol{y}^N(\boldsymbol{x}_t)$ je výstup poslední vrstvy neuronové sítě při vstupu $\boldsymbol{x}_t$ a $\boldsymbol{y}^{*N}(\boldsymbol{x}_t)$ je žádaný výstup (informace od učitele).

Pokud jsou aktivační funkce $g_k^n(\boldsymbol{a}^n)$ neuronové sítě diferencovatelné, iterativní gradientní metody jsou nejpoužívanějšími optimalizačními postupy pro trénování ANN. Jedny z pokročilejší metod trénování urychlující konvergenci trénování jsou např.:

- **Algoritmus BFGS s limitovanou pamětí** (L-BFGS – Limited memory Broyden, Fletcher, Goldfarb and Shanno) - patří do třídy *Kvasi-Newtonových metod*, namísto výpočtu celé matice Hessianu je iterativně počítána pouze její aproximace. Hessian není ukládán do paměti celý, ale jen jeho některé řádky (více lze nalézt v [118]).
- **Algoritmus zlepšené odolné propagace** (IRPROP – Improved Resilient Propagation) - je založený na zlepšení konvergence trénování vyhnutím se přímého výpočtu gradientu. Algoritmus využívá pro výpočet kroku gradientní metody namísto gradientu samotného pouze jeho znaménko k určení směru k minimu funkce [119].

6.7.2 Bottleneck

Jedním z možných využití ANN je mimo jiné komprese dat. Používanou strategií pro kompresi je neuronová síť nazývaná *bottleneck* [120]. Úkolem neuronové sítě je distribuce vstupních dat s dimenzí D (odpovídá počtu neuronů vstupní vrstvy ANN) přes vnitřní vrstvu sítě s nižší redukovanou dimenzí B, výstupní vrstva má pak opět původní dimenzi D. Trénování sítě probíhá na datech, která jsou stejná na vstupu i na výstupu, po natrénování sítě lze výstup její vnitřní vrstvy brát jako výstup kompresoru, zbytek sítě funguje jako dekompresor (viz obrázek 6.5).

V úloze adaptace s malým počtem adaptačních dat však nehledáme bezztrátovou kompresi vstupních dat, naopak chceme dosáhnout redukce nepodstatné nebo chybné informace uložené v chybně odhadnuté adaptaci. Pro takový úkol lze natrénovat ANN bottleneck nikoliv na schodných stupech a výstupech, ale na vstup trénované ANN přivést chybně odhadnutou adaptaci a na výstup pak její správnou variantu (adaptaci odhadnutou na dostatečném množství adaptačních dat). Neuronová síť pak redukuje vliv špatně odhadnutých parametrů adaptace, ale ponechá informaci od parametrů, které byly odhadnuty správně (více v podkapitole 7.8.4).

Kapitola 7

Experimenty, vlastní modifikace adaptačních metod

V této kapitole jsou srovnány výsledky vybraných adaptačních metod popsaných v této práci, které byly programově realizovány. Navržené experimenty byly zaměřeny na adaptaci řečníka jak v supervised, tak i v unsupervised úloze s různým počtem adaptačních dat. Výsledky zde uvedené jsou pouze k porovnání účinnosti jednotlivých adaptačních metod, nikoliv celého systému rozpoznávání řeči.

Do testů je doplněn popis řešení navržených pro zlepšení daných adaptačních postupů. Tyto metody jsou pak porovnány s jejich původními verzemi. Testy metod byly provedeny na dvou korpusech popsaných níže: Českém telefonním korpusu a české části SpeechDat-East korpusu.

7.1 Korpusy a nastavení pro experimenty

7.1.1 Český telefonní (CzT) korpus

Korpus telefonních dat obsahuje nahrávky více než 1300 řečníků. Každý z řečníků byl požádán o to, aby přečetl do telefonu 40 vět s průměrnou délkou 8 sekund. Databáze textů byla vytvořena z elektronických stránek českých novin, a to tak, aby byla foneticky vyvážená vzhledem k relativním výskytům trifónů v přirozeném jazyce [121]. Telefonní nahrávky byly zpracovány kartou DIALOGIC D/21D, a to s využitím vzorkovací frekvence 8 kHz v 8bitovém rozlišení. Takto zvolená vzorkovací frekvence nám podle Nyquistova teorému zaručuje zpracování signálu v pásmu 0 až 4000 Hz, což plně postačuje při práci se signálem telefonní kvality.

Pro účely trénování a testování byly všechny pořízené nahrávky transkribovány pomocí anotačního softwaru Transcriber (`http://www.ldc.upen.edu`). Při tomto procesu je zapsán skutečný text vyslovený řečníkem, včetně případných přeřeků, nedořeků či různých neřečových událostí. Mezi neřečové události můžeme zařadit například hlasitý nádech, 'hlasité přemýšlení' (er, ehm, hm, apod.), mlasknutí jazyka, vzdálený hluk, vzdálenou řeč apod.

Množina telefonních nahrávek byla rozdělena na disjunktní množiny:

- **Trénovací sadu** namluvenou 100 řečníky, kde každý z nich přečetl 40 různých foneticky vyvážených vět. Celkově tedy bylo pro trénování k dispozici 4000 trénovacích vět. Na této sadě byl natrénován SI akustický model.
- **Evaluační sadu** obsahující nahrávky odlišných 76 řečníků, kde každý řečník měl stejnou množinu 20 foneticky vyvážených vět. Ta byla rozdělena vedví na:

- **Adaptační sadu** s 15 větami od každého řečníka z evaluační sady. Za účelem testování různého množství dat pro adaptaci byly vytvořeny skupiny čítající 1 až 12 testovacích vět od každého řečníka.
- **Testovací sadu** odlišných 5 vět od každého řečníka z evaluační sady.

Data byla zparametrizována MFCC parametrizací, 11 dimenzionální vektory pozorování byly získány z 32 ms dlouhého Hamingova okénka s posunem 10 ms. Byla použita kepstrální normalizace CMN a byly přidány Δ, Δ^2 dynamické koeficienty.

Natrénovaný akustický model byl třístavový trifónový HMM s 8 složkami v každém stavu s diagonální kovarianční maticí. Systém ASR neobsahoval žádný jazykový model. Slovník pro přepis obsahoval 475 různých slov, kde několik z nich mělo více různých fonetických přepisů, tedy finální počet položek ve slovníku byl 528. V promluvách se nenacházela žádná **slova mimo slovník** (OOV – Out Of Vocabulary).

7.1.2 SpeechDat-East (SD-E) korpus

SpeechDat-East korpus [122] obsahuje telefonní nahrávky v pěti jazycích (čeština, polština, slovenština, maďarština a ruština). Jednotlivé nahrávky jsou rozděleny po větách, jež mají značnou variabilitu v délce (některé věty mohou být i jednoslovné), průměrně je délka věty pouhé 4 sekundy. Pro naše testování jsme použili českou část, ze které jsme vybrali řečníky do následujících sad:

- **Trénovací sada** obsahuje 700 řečníků s 50 různými větami pro každého z nich.
- **Testovací sada** se skládá z 200 řečníků, pro které bylo opět k dispozici 50 vět. Tyto věty neobsahují referenční přepis, adaptace na nich testovaná je unsupervised, tedy není třeba z této části vyčleňovat věty pro adaptační část, adaptace je provedena na testovacích větách. Pro účely testování různého množství dat pro adaptaci byly vytvořeny skupiny čítající 1 až 12 adaptačních vět od každého řečníka.

Byla provedena MFCC parametrizace akustických dat, 11 dimenzionální vektory pozorovaní byly získány z 32 ms dlouhého Hamingova okénka s posunem 10 ms. Byla použita kepstrální normalizace CMN a přidány Δ, Δ^2 dynamické koeficienty.

Z trénovací sady byl odhadnut třístavový trifónový model s 2105 stavy a 8 složkami s diagonálními kovariančními maticemi. Byl použit trigramový jazykový model [3] a slovník se 7000 slov.

7.2 Hodnocení úspěšnosti rozpoznávání

V úlohách rozpoznávání řeči je výsledný přepis porovnán s referenčním textem dané promluvy pomocí algoritmu dynamického borcení času (DTW – Dynamic Time Warping) [123]. Úspěšnost přepisu se dá hodnotit [35] například pomocí **procenta chybně rozpoznaných slov** (WER – Word Error Rate), **přesností** (Acc – Accuracy) a **správností** (Corr – Correctness) výsledného přepisu. Slovo správně rozpoznáno je označeno jako H (Hit), špatně rozpoznáno S (Substitution), slova, která v přepisu chybí, jsou D (Delete), a ta, která přebývají, I (Inzertion). Jednotlivé míry úspěšnosti lze psát ve tvaru

$$WER = \frac{S + D + I}{N} 100\%, \tag{7.1}$$

$$Acc = 100\% - WER, \tag{7.2}$$

$$Corr = \frac{H}{N}100\%, \tag{7.3}$$

kde N je počet všech slov. V této práci je pro porovnávání používána míra úspěšnosti WER, počítána s pomocí programu *HResults.exe* ze softwarového balíku HTK verze 3.4 [5].

7.3 Statistická významnost experimentů

Při porovnávání různých systémů, testovaných na omezené trénovací množině, není pouhý rozdíl skóre daných systémů plně vypovídající [124]. Pro porovnání systémů jsou obvykle využívány údaje o statistické významnosti dosaženého výsledku. Definujme nulovou hypotézu H_0: dva výsledky různých systémů pocházejí ze stejné pravděpodobnostní distribuce. Hladina statistické významnosti testu je určena pravděpodobností zamítnutí této nulové hypotézy. Pokud je pravděpodobnost H_0 nižší než zvolená hladina významnosti, lze prohlásit, že výsledek je statisticky významný na této hladině významnosti.

Pro odhad distribuční funkce je nutno mít hodnoty výsledků systému pro různé testovací množiny. Při omezeném množství testovacích dat je možné získat odhad distribuční funkce metodou **křížové validace** (cross-validation) [125], kde jsou testovací data několikrát rozdělena do různých testovacích množin. Tento postup vyžaduje několikanásobné testování systému a je tedy časově náročný.

Jinou možností je metoda **bootstrap** [126], kdy je použito převzorkování originálních změřených hodnot pro získání více výsledků testu. Tímto postupem lze odvodit jak statistickou významnost, tak i konfidenční interval testu. Máme-li testovací množinu T_0 s přepsanými N větami, další přepsané testovací množiny o velikosti N pak vytváříme náhodným vybíráním z T_0. Tímto postupem získáme M dalších testovacích množin T_m pro $m = 1 \ldots M$, ze kterých můžeme spočítat parametry distribuční funkce testu. Předpokladem použití této metody je, že použité věty jsou reprezentanty testovaného souboru.

V práci [127] je alternativně použita metoda **aproximativní randomizace** (approximate randomization) pro vyhodnocování statistické významnosti testu, která na rozdíl od metody bootstrap pracuje s výsledky obou porovnávaných systémů. Na začátku máme dvě množiny testů T_0^A a T_0^B provedených na systémech A a B pro stejnou testovací množinu. Definujeme rozdíl ve skóre těchto systémů

$$D_0 = \left| S_0^A - S_0^B \right|. \tag{7.4}$$

Náhodně s pravděpodobností 0,5 prohazujeme výsledky testu jednoho systému každé konkrétní věty za výsledek druhého systému. Tímto postupem získáme další výsledky testované množiny T_m^A a T_m^B. Opakováním tohoto postupu dostaneme M rozdílů ve skóre D_m pro $m = 1 \ldots M$. Pro určení hladiny významnosti testu je nutné spočítat, kolikrát byl nový rozdíl převzorkovaných systémů D_m větší nebo roven původnímu rozdílu D_0, tedy $D_m \geq D_0$. Pokud tento počet označíme C, pravděpodobnost nulové hypotézy (oba systémy mají shodnou distribuční funkci a tedy jejich rozdíl není statisticky významný) je dána

$$P = (C + 1)/(M + 1). \tag{7.5}$$

Nulovou hypotézu můžeme zamítnout na dané hladině významnosti $\bar{P}$, pokud $\bar{P} \geq P$.

V této práci byla použita metoda bootstrap pro výpočet konfidenčního intervalu testu a metoda aproximativní randomizace pro určení hladiny významnosti testu. Tyto údaje jsou uvedeny na konci této kapitoly pro nejlepší porovnávané systémy.

7.4 Klasické metody adaptace

V této části jsou porovnány výsledky klasických metod z kapitoly 3 a to jmenovitě metody MAP (viz podkapitola 3.3) a metody lineárních transformací (MLLRmean, MLLRcov a fMLLR popsané v podkapitole 3.4). Experimenty byly provedeny na korpuse CzT. Rozčlenění výsledků pro jejich vzájemné porovnání vychází z rozdělení adaptačních metod, které bylo popsáno v podkapitole 3.1.

Nastavení metod použitých v těchto testech je následující:

V metodě MAP byly adaptovány střední hodnoty, kovarianční matice i váhy složek najednou. Konstanta τ byla experimentálně nastavena na hodnotu 16 (výsledky pro různá nastavení τ, viz tabulka A.1 v Přílohách).

Regresní strom v metodě MLLR (resp. fMLLR) byl konstruován pomocí HTK verze 3.4 [5]. Ke konstrukci byla využita pouze blízkost středních hodnot HMM v akustickém prostoru. Strom měl 32 listových uzlů, tedy 32 základních shluků, které se pak podle aktuálního množství adaptačních dat spojovaly do sebe dle navrženého stromu. Okupační práh třídy regresního stromu byl zvolen $Th = 1000$. Počet vnitřních iterací pro výpočet transformační matice byl fixován na hodnotě 20 (viz tabulka A.2 v Přílohách). SI označuje neadaptovaný model.

7.4.1 Transformace modelu vs. transformace vektoru pozorování

V prvním řádku tabulky 7.1 jsou uvedené hodnoty Acc pro experiment s SI modelem a modely adaptovanými metodami MAP, MLLRmean, MLLRcov, fMLLR. Veškeré výsledky jsou získány pouze po jedné adaptační iteraci, výjimkou je metoda MLLRcov, která je z principu dvouiterační (viz podkapitola 3.4.1). Obecně lze předpokládat další zlepšování pro více adaptačních cyklů (viz tabulka A.3 v Přílohách), což není pro porovnání jednotlivých metod signifikantní. Druhý řádek tabulky uvádí průměrný čas výpočtu adaptace na jednoho řečníka[1].

Tabulka 7.1: Výsledky (Acc[%]) vybraných adaptačních metod a trvání jejich odhadu [s], pro korpus CzT.

	SI model	MAP	MLLR mean	MLLR cov	fMLLR
Acc	65,32	73,09	75,01	**77,93**	76,94
čas adaptace		**0,69**	2,39	17,03	14,91

Z výsledků je vidět podstatné zlepšení přesnosti rozpoznávání při použití adaptačních metod, a to až o 17 % relativně vůči SI modelu. Nejlepší výsledky dává metoda MLLRcov, která adaptuje jak střední hodnoty, tak kovarianční matice modelu, a to různými transformacemi. Tato metoda v přesnosti předčí i fMLLR (ta adaptuje střední hodnoty a kovarianční matice stejnou transformací), ale je ze všech testovaných metod nejpomalejší.

Nejrychlejší metodou je MAP, pro kterou jsme v testu měli dostatečné množství dat, proto i ona má dobré výsledky. Malou rychlost adaptací založených na lineárních transformacích lze přičítat hlavně velkému regresnímu stromu. Pro takové množství adaptačních dat obsažených v našem testu bylo vytvořeno v průměru 10 transformačních matic pro každého řečníka. Rapidní

[1] Výpočet prováděn na domácí stanici s procesorem Core2duo a vnitřní pamětí 2 MB.

zpomalení metod založených na lineární transformaci je způsobeno nutností iteračního výpočtu matic uvnitř algoritmu adaptace. Naopak podstatnou výhodou metody fMLLR je její aplikace na vektory pozorování, tedy metoda obchází nutnost načítání velkého množství parametrů celého nového akustického modelu pro každého řečníka. Místo toho transformuje rozpoznávaná data pomocí afinní transformace.

Obecně dosažené výsledky podporují teoretické předpoklady těchto metod uvedených v kapitole 3. Pro další testování byla upřednostněna metoda fMLLR, díky výše jmenovaným pozitivům, před ostatními adaptačními metodami.

7.4.2 Diskriminativní vs. generativní adaptace

Porovnání výsledků při rozdílných přístupech k adaptaci založené na generativním a diskriminativním kritériu lze vidět v tabulce 7.2. Dvě rozdílené metody (MAP a (f)MLLR) jsou zde porovnány ve variantě generativní a diskriminativní (s přívlastkem D ve jméně metody). Popis diskriminativního přístupu k adaptaci lze najít v podkapitolách 3.3.1 pro DMAP přístup a 3.4.3 pro D(f)MLLR. Vážící faktor f definující brzdící faktor v DMAP a D(f)MLLR adaptaci byl zvolen roven 1. Pro lineární transformace byla odhadována pouze globální transformace.

Tabulka 7.2: Výsledky (Acc[%]) metod MAP a MLLR při použití generatiního a diskriminativního přístupu, pro korpus CzT.

MAP	DMAP	MLLR	DMLLR	fMLLR	DfMLLR
73.09	75,41	75,01	74,04	76,94	77,02

Výsledky ukazují mírné zlepšení při použití diskriminativních metod adaptace, ačkoli toto zlepšení je omezeno množstvím dat pro adaptaci. Diskriminativní kritéria byla původně odvozena pro metody trénování (viz kapitola 2.3.3) a již v těchto podmínkách prokázala vyšší potřebu trénovacích dat oproti generativním přístupům.

7.4.3 Inkrementální vs. dávková adaptace

V tabulce 7.3 je porovnání výsledků pro metodu fMLLR v inkrementálním a v dávkovém režimu, a to jak pro globální transformaci, tak s využitím regresního stromu (RT – Regresion Tree) pro okupační práh $Th = 1000$. Uvedené výsledky jsou pro korpus CzT. V inkrementálním režimu byla provedena adaptace po každé adaptační větě na rozdíl od dávkového režimu, kdy výpočet adaptace proběhl až po nasčítání statistik od všech adaptačních vět. Nastavení metody fMLLR (především volba RT) je uvedeno na začátku této podkapitoly.

Inkrementální adaptace je převážně využívána při on-line adaptaci, kdy adaptujeme testovaného řečníka v průběhu rozpoznávacího procesu. Nebyla pro něj tedy dostupná žádná data před jeho vlastním rozpoznáváním. Výsledky tohoto přístupu jsou lepší než pro dávkovou adaptaci. To proto, že adaptace je provedena několikrát, vždy při příchodu další adaptační věty, aniž by předchozí statistiky byli zapomenuty. Každým krokem je tedy odhad transformační matice zpřesňován (další iterací navíc s větším množstvím informace). Poznámka: pro testování bylo použito pro každého řečníka 15 vět, výsledky inkrementální adaptace vzdáleně simulují proces

Tabulka 7.3: Výsledky (Acc[%]) pro metodu fMLLR s globální tranformační maticí nebo s regresním stromem, pro inkrementální a dávkový přístup, pro korpus CzT.

	inkrementální fMLLR	dávková fMLLR
global	76,71	75,45
RT1000	76,90	76,94

odpovídající 15 iteracím (s postupným navyšováním adaptačních dat).

7.4.4 Unsupervised Adaptace

V tabulce 7.4 lze porovnat výsledky dvou alternativ adaptace fMLLR, s dostupnými referenčními přepisy (supervised) a bez nich (unsupervised). V unsupervised případě je přepis adaptačních dat získán z jednoho průchodu ASR systému, tedy obsahuje chyby rozpoznání, proto je zde využita informace o jistotě rozpoznaného slova, tzv. CF (viz podkapitola 5.1.1).

Tabulka 7.4: Výsledky (Acc[%]) pro metodu fMLLR s globální tranformační maticí nebo s regresním stromem pro okupační práh $Th = 1000$, pro supervised a unsupervised variantu s využitím informace o CF u rozpoznaných i okolních slov, pro korpus CzT.

	SI	supervised fMLLR	unsupervised fMLLR
global	65,32	75,45	71,20
RT1000	65,32	76,94	70,80

Metody unsupervised adaptace vykazují očekávané snížení úspěšnosti při rozpoznávání. Příčinou je nižší přesnost přepisu adaptačních dat, a to i když uvažujeme pouze jisté přepisy, tedy přepsaná slova s CF $> 0,98$. S využitím CF také souvisí nižší počet adaptačních dat (některá jsou kvůli nízké věrohodnosti přepisu nevyužita).

7.4.5 Adaptační trénování

Tabulka 7.5 obsahuje výsledky neadaptovaného SI modelu a modelů získaných adaptačním trénováním (viz kapitola 4, jmenovitě metoda SAT a metoda VTLN).

Metoda SAT přetrénovala střední hodnoty i kovarianční matice modelu a byla založena na fMLLR transformacích z podkapitoly 4.1.2. Nastavení metody fMLLR bylo stejné jako v prvním experimentu popsaném výše v podkapitole 7.4, tedy regresní strom s 32 listovými uzly a $Th = 1000$.

Metoda VTLN přetrénovala také střední hodnoty i kovarianční matice a byla založena na lineárních transformacích (VTLN-LT viz podkapitola 4.4). I pro odvození VTLN-LT transformací byl využit regresní strom, ale s 64 listovými uzly a okupačním prahem nastaveným na $Th = 100$. Adaptační metoda založená na VTLN odhaduje pouze jeden parametr α pro každou třídu, nepotřebuje tedy tak velké množství adaptačních dat. Jako warpovací funkce byla

zvolena po částech lineární funkce z podkapitoly 4.3.1.

Tabulka 7.5: Výsledky rozpoznávání (Acc[%]) systému s neadaptovaným SI modelem a modely SAT a VTLN - vytvořené technikami tzv. adaptačního trénování, pro korpus CzT. Pro porovnání přidánu i výsledky SI modelu adaptovaného metodou fMLLR.

SI	fMLLR	SAT	VTLN
65,32	76,94	**78,21**	71,72

Metody adaptačních technik pro trénování odstraňují z modelu informaci o řečníkovi, model se pak stává vhodnější pro adaptaci a adaptační metody na něm vykazují lepší účinnost v porovnání s SI modelem. Metoda SAT v experimentech dokázala své opodstatnění, oproti samotné adaptaci pouze na SI model vykázala cca 2 % relativního zlepšení.

7.5 Kombinace adaptačních metod

Výhodou metody MAP je fakt, že při dostatečném množství dat SA model konverguje k SD modelu. Naopak výhodou metody fMLLR je její dobrá účinnost i při malém počtu adaptačních dat díky shlukování podobných složek hustotních směsí, a tím snižování počtu volných parametrů modelu. Další výhodou je její aplikace přímo na vektory pozorování. Nabízí se možnost výše zmíněné metody kombinovat dohromady.

7.5.1 Dvoukroková adaptace

Jednou z možností, která se intuitivně jeví jako nejjednodušší, je adaptace modelu ve dvou krocích [128]. S ohledem na princip metody (f)MLLR a MAP je výhodný postup (viz obrázek 7.1):

- 1.krok - Adaptovat SI model pomoci (f)MLLR adaptace, získáme $\mathrm{SD}_{\mathrm{(f)MLLR}}$ model. Pokud není dostatek dat pro adaptaci každého parametru, budou se pomocí (f)MLLR adaptovat společně parametry nashlukované regresním stromem. Adaptační postup lze schématicky vyjádřit zápisem

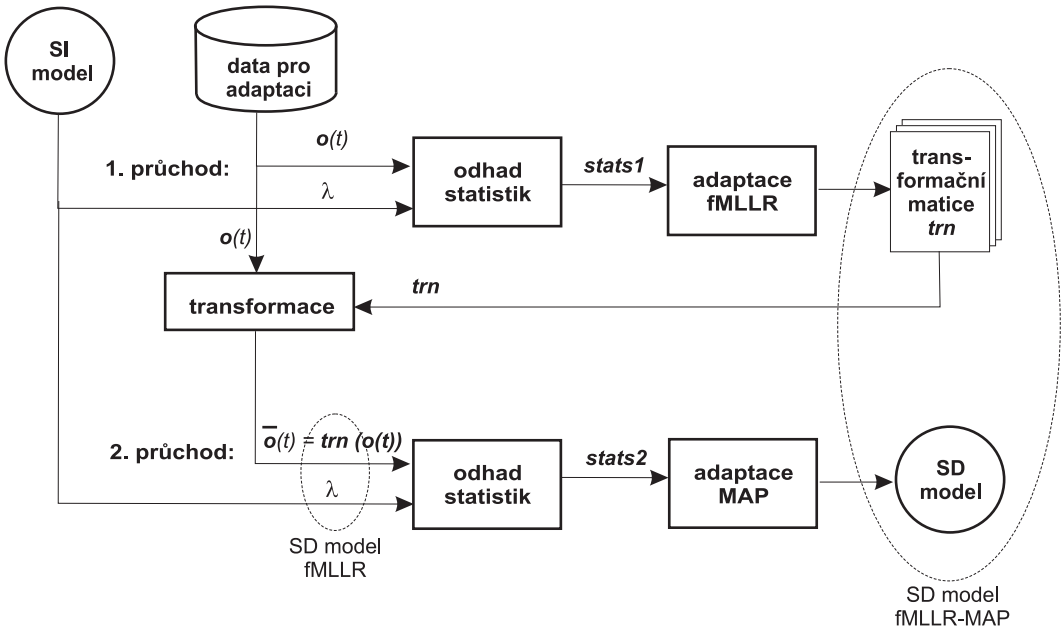
$$\mathrm{SI} \rightarrow \mathrm{stats}_1 \text{ pro SI } \stackrel{\mathrm{(f)MLLR}}{\Longrightarrow} \mathrm{SD}_{\mathrm{(f)MLLR}} \,. \tag{7.6}$$

- 2.krok - Adaptovat $\mathrm{SD}_{\mathrm{(f)MLLR}}$ model pomocí MAP adaptace, získáme $\mathrm{SD}_{\mathrm{(f)MLLR\text{-}MAP}}$ model. Metoda MAP provede adaptaci (zpřesnění) jednotlivých složek, pro které máme dostatečné množství dat. Schématický zápis je následující:

$$\mathrm{SD}_{\mathrm{(f)MLLR}} \rightarrow \mathrm{stats}_2 \text{ pro } \mathrm{SD}_{\mathrm{(f)MLLR}} \stackrel{\mathrm{MAP}}{\Longrightarrow} \mathrm{SD}_{\mathrm{(f)MLLR\text{-}MAP}} \,. \tag{7.7}$$

Druhou možností je aplikovat obě metody v opačném pořadí, výsledky ale nedosahují takového zlepšení (viz výsledky z tabulky 7.6). Metoda (f)MLLR může totiž v druhém kroku posunout i složky, které byly již dobře adaptovány v prvním kroku metodou MAP.

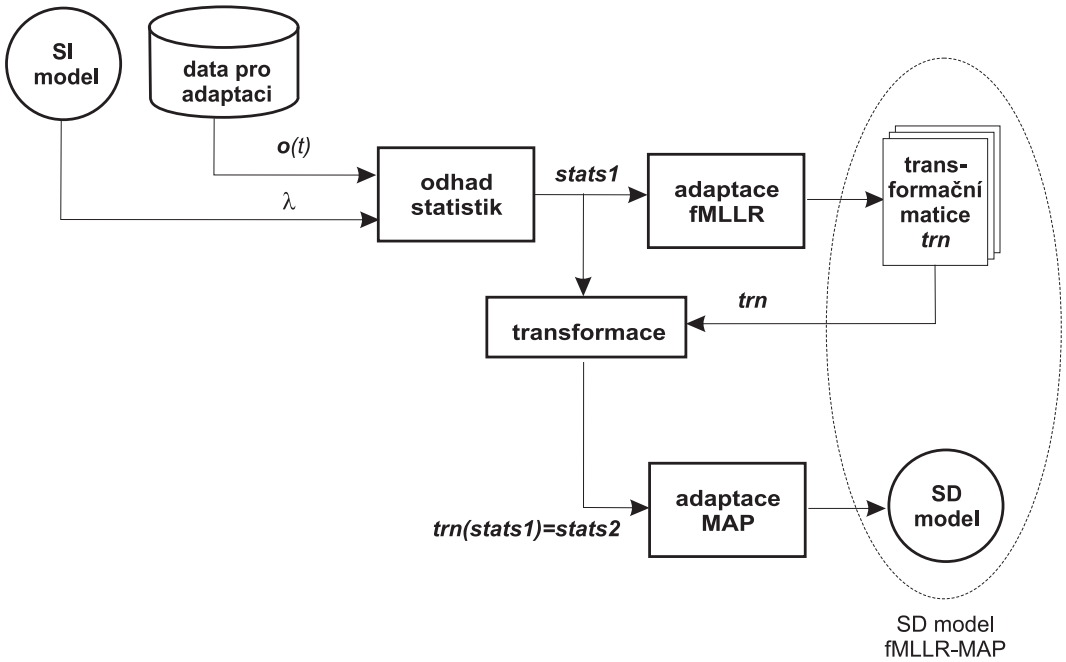
Nevýhodou tohoto dvoukrokového přístupu je jeho časová náročnost, je totiž nutné nasčítávat statistiky adaptačních dat dvakrát, nejprve pro SI model a posléze pro $\mathrm{SD}_{\mathrm{(f)MLLR}}$ model.



Obrázek 7.1: Dvoukroková kombinace fMLLR a MAP adaptace.

7.5.2 Jednokroková adaptace

K výpočtu adaptace MAP i fMLLR se využívají stejné adaptační statistiky, jejichž akumulace je z celého procesu adaptace časově nejnáročnější, proto jsme navrhli kombinaci těchto nejpoužívanějších metod pouze s jedním průchodem adaptačními daty [129].



Obrázek 7.2: Jednokroková kombinace fMLLR a MAP adaptace.

Návrh s výhodou využívá vlastnosti metody fMLLR, kdy vypočtená adaptace je aplikována formou transformačních matic na vektory příznaků pozorování. Stejná transformace lze však také aplikovat přímo na již akumulované statistiky. Namísto transformace všech adaptačních dat pro výpočet adaptace druhého kroku lze tedy transformovat pouze již nasčítané statistiky, a tím se vyhnout časově náročnému procesu akumulace statistik pro nově adaptovaný model.

Postup jednokrokové kombinace spočívá nejprve v nasčítání statistik adaptačních dat pro SI model stats_1, z nichž je metodou fMLLR vypočítána adaptační matice $\boldsymbol{A}$,$\boldsymbol{b}$. Stávající statistiky jsou poté fMLLR adaptací (matice $\boldsymbol{A}$,$\boldsymbol{b}$) transformovány do nového akustického prostoru (viz obrázek 7.2):

$$\bar{\boldsymbol{\varepsilon}}_{jm}(\boldsymbol{o}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)(\boldsymbol{A}_{(n)}\boldsymbol{o}(t) + \boldsymbol{b}_{(n)})}{\sum_{t=1}^{T} \gamma_{jm}(t)} = \boldsymbol{A}_{(n)}\boldsymbol{\varepsilon}_{jm} + \boldsymbol{b}_{(n)} \,, \tag{7.8}$$

$$\begin{aligned} \bar{\boldsymbol{\varepsilon}}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) &= \frac{\sum_{t=1}^{T} \gamma_{jm}(t)(\boldsymbol{A}_{(n)}\boldsymbol{o}(t) + \boldsymbol{b}_{(n)})(\boldsymbol{A}_{(n)}\boldsymbol{o}(t) + \boldsymbol{b}_{(n)})^{\mathrm{T}}}{\sum_{t=1}^{T} \gamma_{jm}(t)} = \\ &= \boldsymbol{A}_{(n)}\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})\boldsymbol{A}_{(n)}^{\mathrm{T}} + 2\boldsymbol{A}_{(n)}\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o})\boldsymbol{b}_{(n)}^{\mathrm{T}} + \boldsymbol{b}_{(n)}\boldsymbol{b}_{(n)}^{\mathrm{T}} \,, \end{aligned} \tag{7.9}$$

kde $\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o})$, $\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})$, $\gamma_{jm}(t)$ jsou prvotní statistiky akumulované SI modelem a $\bar{\boldsymbol{\varepsilon}}_{jm}(\boldsymbol{o})$, $\bar{\boldsymbol{\varepsilon}}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})$ odpovídají statistikám pro model s fMLLR transformačními maticemi. Statistiky $\gamma_{jm}(t)$ nelze jednoduše transformovat do nového akustického prostoru, zůstávají nezměněné.

S pomocí transformovaných statistik lze vypočítat druhou adaptaci MAP, která již adaptuje přímo akustický model. Výsledný na řečníku závislý SD model je dán novým MAP modelem a fMLLR transformacemi. Celý adaptační postup lze schématicky vyjádřit zápisem

$$\text{SI} \rightarrow \text{stats}_1 \overset{\text{fMLLR}}{\Longrightarrow} \text{SD}_{\text{fMLLR}} \rightarrow \text{transformace stats}_1 \rightarrow \text{stats}_2 \overset{\text{MAP}}{\Longrightarrow} \text{SD}_{\text{fMLLR-MAP}} \,. \tag{7.10}$$

I přesto, že $\gamma_{jm}(t)$ zůstávají adaptací fMLLR nedotčené, transformované statistiky nejsou tímto zjevným nedostatkem ovlivněny, jak dokazují výsledky (viz tabulka 7.6).

7.5.3 Porovnání kombinačních přístupů MAP a (f)MLLR

Tabulka 7.6 zobrazuje výsledné Acc systému rozpoznávání získané po kombinaci vybraných metod adaptace otestovaných na korpusu CzT. Kombinace spočívala v postupné adaptaci pomocí dvou různých metod aplikovaných v jednokrokové nebo dvoukrokové variantě. Byly použity adaptační metody se stejným nastavením popsaným v podkapitole 7.4 (regresní strom s 32 listovými uzly a $Th = 1000$).

Tabulka 7.6: Výsledky (Acc[%]) kombinace adaptačních metod MAP, MLLR a fMLLR pro korpus CzT.

MAP -MLLRmean	MAP -MLLRcov	MAP -fMLLR	MLLRmean -MAP	fMLLR -MAP(dvoukrok)	fMLLR -MAP(jednokrok)
77,03	78,14	78,37	77,22	**79,51**	78,84

Kombinace metod vykazují další zlepšení adaptace. Optimálním z hlediska účinnosti se jeví kombinace metody fMLLR a MAP (aplikované v tomto pořadí). Dvoukroková metoda

fMLLR-MAP je z testovaných kombinací tou nejúčinější. Navrhovaná jednokroková varianta této metody se jí účinností skoro vyrovná, je však časově podstatně méně náročná (viz kapitola 7.5.2).

7.5.4 Porovnání kombinace přístupů DMAP a DfMLLR

Jednokrokový postup kombinace metod fMLLR a MAP se dá stejným způsobem využít i pro jejich diskriminativní verze DfMLLR a DMAP publikované v [130]. Vzrůstá však časová náročnost adaptace, protože je potřeba transformovat více statistik (viz 3.2). Výsledky této kombinace pro korpus CzT (v porovnání s dvoukrokovou kombinací) lze nahlédnout v tabulce 7.7.

Tabulka 7.7: Výsledky (Acc[%]) kombinace adaptačních metod DMAP a DfMLLR pro korpus CzT.

DfMLLR -DMAP(dvoukrok)	DfMLLR -DMAP(jednokrok)
79,61	79,44

Opět jako v nediskriminativním případě je dvoukroková kombinace účinnější, ale jednokrokový přístup vynechává nutnost opětovného akumulování adaptačních statistik, a tím podstatně zrychluje adaptaci.

7.6 On-line adaptace

Dílčí problémy on-line přístupu k adaptaci (inkrementální adaptace, unsupervised adaptace), popsané v kapitole 5, již byly otestovány v podkapitolách 7.4.3 a 7.4.4. Zde je uveden experiment na celém on-line systému pro rozpoznávání testovaný na reálných datech z Poslanecké sněmovny Parlamentu České republiky.

7.6.1 Popis experimentu

On-line systém pro titulkování přímých přenosů z Poslanecké sněmovny Parlamentu České republiky vysílaných Českou televizí [131] byl využit pro testování on-line adaptace. Akustický model (třístavový HMM s 8 složkami GMM pro každý stav) byl natrénován na 100 hodinách nahraných z přímého přenosu z Parlamentu České republiky s manuálně přepsanými daty. Dodatečně bylo provedeno disktiminativní dotrénování HMM.

Analogový vstupní signál byl zdigitalizován při vzorkování 44.1 kHz a v 16bitovém rozlišení. PLP parametrizace obsahovala 19 filtrů a 12 PLP cepstralních koeficientů s Δ a Δ^2 dynamickými koeficienty.

Jazykový model (LM - Language Model) byl natrénován s cca 24M tokeny Good-Turing algoritmem pomocí SRI Language Modeling Toolkit [132]. Slovník obsahoval 177 125 slov. Pro

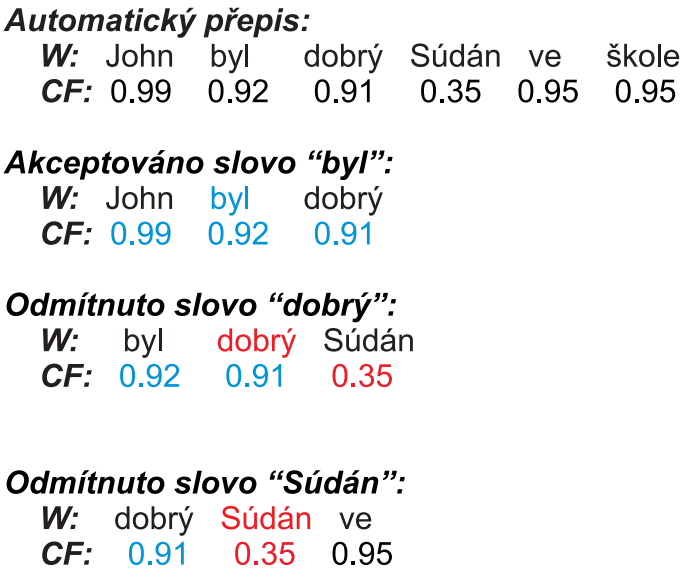
rychlé on-line rozpoznávání byl použit bigramový LM, pro větší přesnost přepsaných slov pak trigramový LM.

Experimenty byly provedeny na 12 nahrávkách od různých řečníků, každý s délkou 5 minut. Jako adaptační metoda byla zvolena metoda fMLLR využívající informaci z regresního stromu a okupační práh $Th = 1000$. Adaptace probíhala inkrementálně vždy pro určité kvantum testovacích dat (která byla před tím přepsána systémem). Ideálně by mohla být adaptační matice přepočítávána po každém nově přepsaném slově, ale to by bylo v úloze on-line rozpoznávání časově náročné. Proto byl zvolen práh $T = 1000$ nových dat, kdy byla adaptační matice znovu přepočítána.

7.6.2 Informace o jistotě rozpoznávání

V on-line rozpoznávání nemáme referenční přepis k adaptačním datům, je tedy nutno využít přepisu získaného prvním průchodem ASR systému. Tento přepis není bezchybný, proto je zde využita informace o jistotě rozpoznaného slova, tzv. CF přepisu. CF popsaný v podkapitole 5.1.1 je ohodnocení připadající jednotlivým slovům, neměří však přesnost hranice mezi přepsanými slovy. Stále tak může docházet k chybám, protože hranice správně rozpoznaných slov nemusí být určeny bezchybně díky nepřesnému přepisu jejich sousedních slov. V práci [68] jsme navrhli postup, jak tomuto nepříznivému stavu zabránit. Pro výpočet adaptace je brán v úvahu také CF levého a pravého kontextu uvažovaného slova (viz příklad na obrázku 7.3). Pro výpočet adaptace akceptujeme jen data/slova, která splňují současně obě následující podmínky:

1. jejich přepis W je dostatečně přesný, $CF_W > T_{CF}$, kde T_{CF} je apriori volený práh.
2. přepis jejich sousedních slov $W^{\pm 1}$ je také dostatečně přesný, $CF_{W^{\pm 1}} > T_{CF}$.



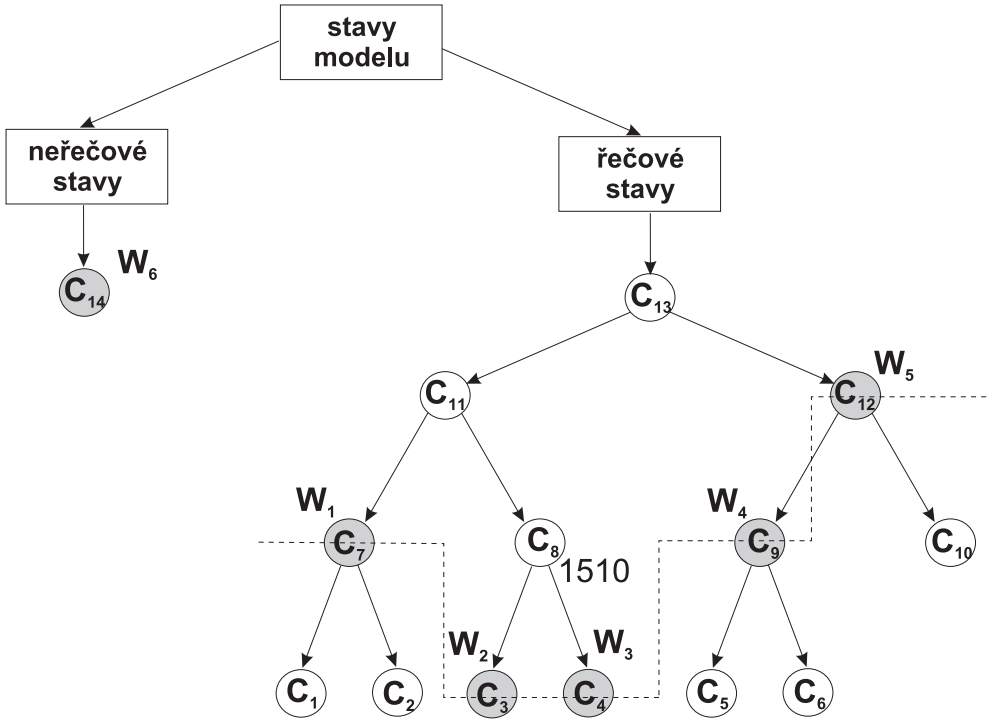
Obrázek 7.3: Ilustrační příklad automatického přepisu W s přiděleným faktorem jistoty CF. S ohledem na zvolený práh $T_{CF} = 0,9$, bude slovo 'byl' akceptováno pro adaptaci, avšak slova 'dobrý' a 'Súdán' již nikoliv (v závislosti na jejich CF, resp. na CF jejich kontextu).

Tento postup sice sníží počet adaptačních dat, avšak jejich přepis se blíží k referenčnímu přepisu, což je náš hlavní cíl.

7.6.3 Adaptace neřečových událostí

Pokud je použit regresní strom (RT) při určování tříd pro metodu fMLLR, řeč i neřečové segmenty promluvy mohou být zařazeny do stejné třídy RT (jsou adaptovány stejnou transformační maticí). V případech, kdy adaptační data obsahují pouze malé množství neřečových událostí, může dojít k nežádoucí adaptaci stavů HMM odpovídajících těmto neřečovým událostem směrem k řečovým datům. Potom mohou být neřečové události (nádech, odkašlání, mumlání, ...) chybně rozpoznány jako řeč. To může nastat, pokud se významně liší kanál trénovacích dat původního SI modelu od kanálu aktuálně adaptovaného řečníka.

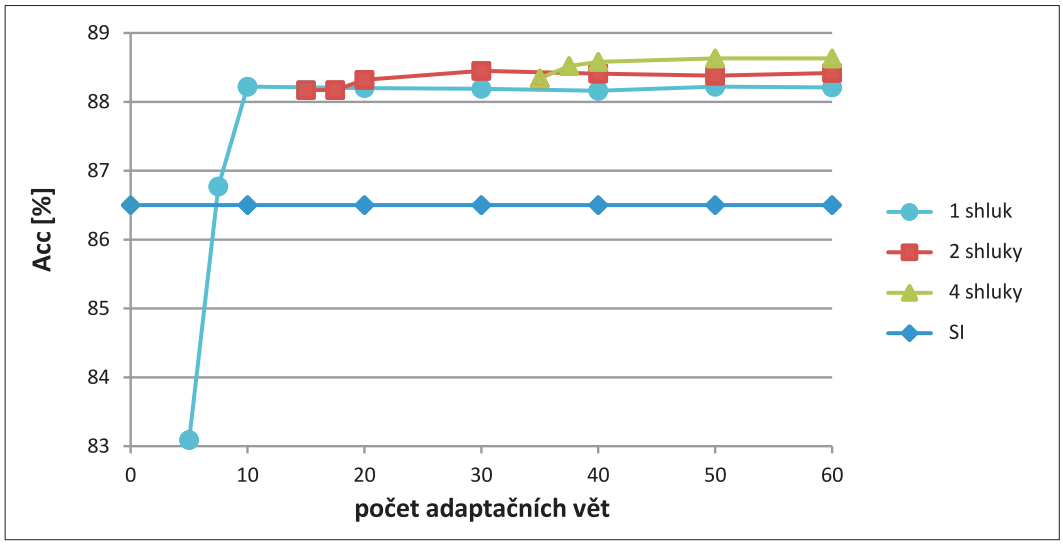
Obecně lze říci, že řeč a neřečové události jsou natolik odlišné, že je výhodné je adaptovat jinou transformační maticí. Proto byl v práci [68] do regresního stromu přidán zvláštní uzel jen pro tyto neřečové události (viz obr. 7.4). S tímto uzlem je zacházeno odlišně než se zbytkem RT. Pokud není obsazen dostatečným množstvím adaptačních dat, adaptační matice se nepočítá a neřečové události zůstávají neadaptovány. Tedy nepoužije se pro jejich adaptaci transformační matice nadřazeného uzlu, jako pro ostatní uzly v RT, viz 3.4.4.



Obrázek 7.4: Příklad binárního regresního stromu s uzlem pro neřečové události.

7.6.4 Výsledky on-line adaptace

Výsledky on-line testování (uveřejněné v práci [133]) lze nalézt v grafu 7.5. Individuální iterace fMLLR adaptace jsou vykonány až při nakumulování dostatečného množství adaptačních statistik (tyto body jsou v grafu označeny zvýšením počtu shluků). Počet shluků uvedených v grafu odpovídá obsazeným shlukům v regresním stromu a tedy i počtu odhadovaných transformací. Zlepšení úspěšnosti rozpoznávání po třetí iteraci adaptace oproti SI modelu bylo cca 3 % relativně. Je důležité poznamenat, že reálná délka testovaných dat je přibližně dvakrát větší, než délka adaptačních dat deklarovaná v grafu. Důvody jsou nízké CF některých slov a jejich okolí a neřečové události, na které systém adaptován nebyl.

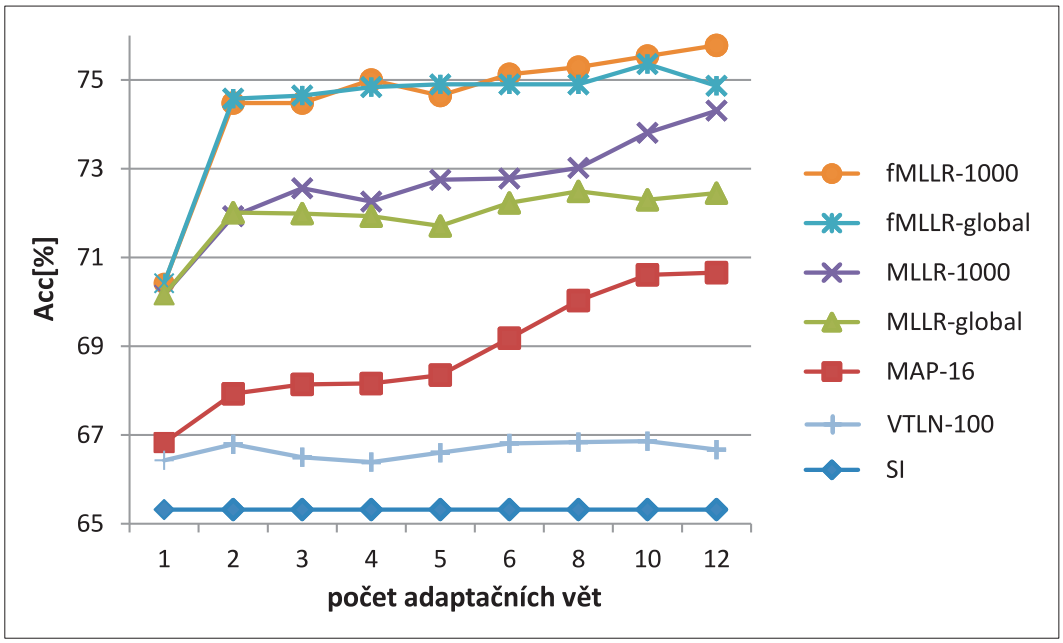


Obrázek 7.5: Výsledky (Acc[%]) on-line adaptovaného systému pomocí metody fMLLR s různým počtem transformací (shluků) na parlamentních datech. SI označuje výsledky modelu bez adaptace.

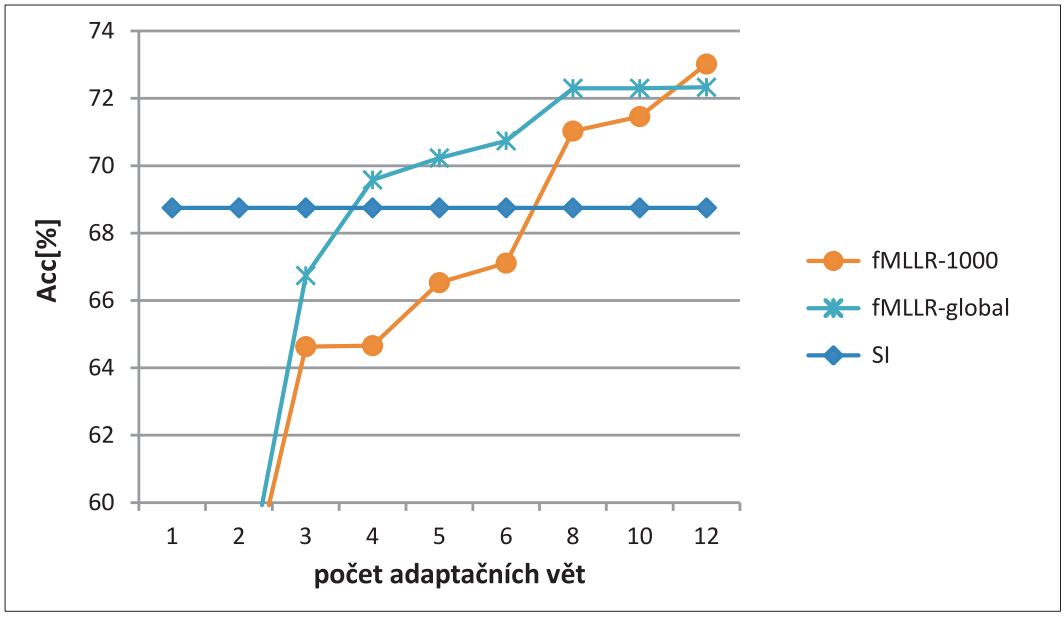
7.7 Množství dat pro adaptaci

Úspěšnost rozpoznávání v závislosti na počtu dat použitých pro adaptaci metodou MAP, (f)MLLR a VTLN je uvedena v grafu 7.6, příslušná tabulka B.1 lze dohledat v Přílohách. Výsledky Acc jsou dány pro různý počet adaptačních vět z korpusu CzT. Označení (f)MLLR-1000 určuje adaptaci (f)MLLR s regresním stromem s prahem $Th = 1000$ a (f)MLLR-global pak pouze globální adaptaci bez regresního stromu, VTLN-100 je adaptace využívající regresní strom s 64 listovými uzly a s prahem $Th = 100$, MAP-16 je adaptace s $\tau = 16$ a SI označuje neadaptovaný model. Průměrná věta pro adaptaci je dlouhá cca 10 s.

Metody založené na lineárních transformacích dokázaly (oproti MAP) adaptovat model již při malém počtu adaptačních dat díky shlukování podobných parametrů modelu. Naopak metoda MAP nabývá na důležitosti s přibýváním adaptačních dat, což jí umožňuje adaptovat více parametrů SI modelu.



Obrázek 7.6: Výsledky (Acc[%]) adaptačních metod při různém počtu adaptačních vět pro korpus CzT.



Obrázek 7.7: Výsledky (Acc[%]) adaptačních metod při různém počtu adaptačních vět pro korpus SD-E.

Korpus CzT obsahuje podstatně delší věty, než aby se ukázal rozdíl mezi metodami (f)MLLR využívající regresní strom a nebo pouze globální transformaci. Z toho důvodu byly provedeny experimenty na korpusu SD-E, který je rozdělen do vět podstatně kratších (4 s i méně) a obsahuje spontánní promluvu bez referenčního přepisu, což ústí v menší počet použitelných dat

pro adaptaci. Výsledky experimentu na korpusu SD-E v závislosti na počtu adaptačních vět jsou zobrazeny v grafu 7.7, tomu odpovídá tabulka B.2 v Přílohách. Označení je schodné jako u grafu 7.6.

Z výsledků experimentů na kratších větách (graf 7.7 pro korpus SD-E) je vidět selhávání metod založených na lineárních transformacích (jak s využitím regresního stromu, tak i jen s globální transformací) pro malé množství adaptačních dat (méně než 6 vět). I když tyto metody dávají dobré výsledky pro adaptaci s dostatečným množstvím dat, je třeba dalšího ošetření těchto metod pro adaptaci s extrémně malým množstvím dat, kdy se odhad transformačních matic stává nestabilním. Tyto problémy jsou řešeny v podkapitole 7.8.

7.8 Robustní přístupy

V této části jsou uvedeny výsledky metod zaměřených na malé množství adaptačních dat zdokumentované v kapitole 6. Dále jsou zde popsány vlastní inovace těchto přístupů a jejich výsledky porovnány s již známými metodami, především pak s metodou fMLLR, která v předchozích experimentech prokázala své výhody. S ohledem na výsledky v podkapitole 7.7 byl pro testování zvolen korpus SD-E.

7.8.1 Zrobustnění statistik

Nejpoužívanější metody adaptace (MAP a LT) a jejich variace využívají ke svým výpočtům statistiky adaptačních dat. Pro řádnou akumulaci těchto statistik je potřeba mít data korektně zarovnaná do jednotlivých stavů akustického modelu, tzv. *force alignment*. I když je k datům dostupný referenční přepis (*supervised adaptation*), zarovnání může obsahovat chyby, způsobené například nevhodným akustickým modelem (ML training nemusí být nejvhodnější odhad HMM [11]). Při *unsupervised* adaptaci je pak zarovnání obvykle ještě nepřesnější, což je dáno nekorektním přepisem způsobeným chybami v ASR (více v kapitole 5.1).

V článku [134] jsme navrhli několik postupů, jak omezit výběr statistik pro vlastní výpočet adaptace. Jednou z nich je vyloučit z výpočtu adaptace statistiky příslušné složky stavu HMM na základě velikosti jejího obsazení c_{jm}, dané rovnicí (3.3). Např. pro metody adaptace založené na lineárních transformacích nebude vyloučená složka uvažována při akumulaci statistiky $\boldsymbol{G}$ a $\boldsymbol{k}$ (viz rovnice (3.39), (3.38)).

Takový přístup může nicméně vyloučit složky stavu s dobře zarovnanými daty, proto je vhodnější posuzovat adaptační data jednotlivě po vektorech pozorování a neakumulovat ty s nízkou hodnotou $\gamma_{jm}(t)$ (3.2). Navrhli jsme dva přístupy, jak vyloučit vektor pozorování z procesu akumulace statistik podle velikosti $\gamma_{jm}(t)$:

- První z možností je stanovit práh Th_γ a brát v úvahu pouze ty vektory pozorování, jejichž $\gamma_{jm}(t) > Th_\gamma$. Takovýto přístup reflektuje předpoklad dvou hypotéz, H_0 a H_1, kde
  H_0: *vektor pozorování* $o(t)$ *BYL generován stavem* j *složkou* m *akustického modelu*,
  H_1: *vektor pozorování* $o(t)$ *NEBYL generován stavem* j *složkou* m *akustického modelu*,
  my chceme minimalizovat nesprávné zavrhnutí hypotézy H_0.

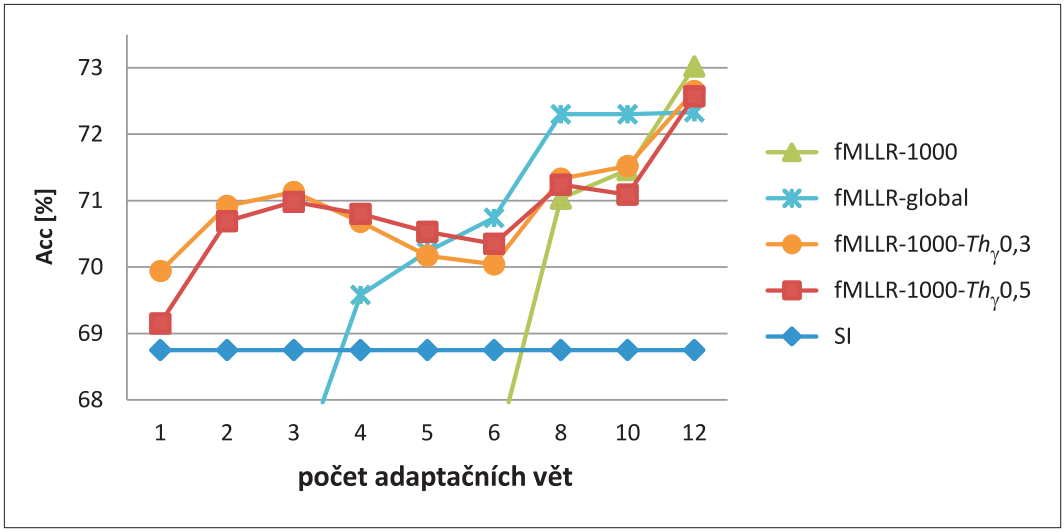
- Druhá možnost spočívá v akumulování pouze statistik přiřazených k N nejlepším složkám daného stavu (s respektem k velikosti $\gamma_{jm}(t)$).

Obě metody jsou v zásadě velmi společné, obě kontrolují (pro konkrétní $o(t)$) počet složek akustického modelu zapojených do akumulace statistik. První pomocí stanovení Th_γ, druhá

pak pomocí N - volený počet nejlepších složek. Jestliže práh Th_γ je nastaven dostatečně vysoký, pak pouze jedna ze složek stavu je akceptována. To odpovídá nastavení $N = 1$ nebo $Th_\gamma > 0,5$. Snížení Th_γ je porovnatelné se zvýšením N. Nicméně Th_γ vyhodnocuje počet akceptovatelných složek s ohledem na konkrétní vektor pozorování.

Výsledky pro robustní statistiky

Výsledky pro různé nastavení prahu ($Th_\gamma = 0,5$ nebo $0,3$) pro metodu fMLLR využívající regresní strom s 32 listovými uzly a $Th = 1000$ lze nalézt v grafu 7.8 nebo též v tabulce B.3 v Přílohách. Označení fMLLR-1000 určuje adaptaci fMLLR s regresním stromem s prahem $Th = 1000$ a fMLLR-global pak pouze globální adaptaci bez regresního stromu. fMLLR-1000-Th_γ označuje fMLLR adaptaci s regresním stromem s prahem $Th = 1000$ pouze s robustními statistikami odpovídající prahu Th_γ. SI označuje neadaptovaný akustický model.



Obrázek 7.8: Výsledky (Acc[%]) adaptace fMLLR s různou volbou prahu Th_γ pro relevanci adaptačních statistik testovaných na SD-E korpusu. Pro porovnání uvedeny i výsledky samotné adaptace fMLLR (globální i s regresním stromem s $Th = 1000$) a výsledky neadaptovaného SI modelu.

Z výsledků je viditelné zlepšení rozpoznávání při výběru adaptačních statistik s respektováním prahu Th_γ. Pro nižší počet dat bylo zlepšení znatelnější, protože v tomto případě je systém citlivější na chybně zarovnaná data. Z výsledků je také patrné, že nedošlo k znatelnému poklesu úspěšnosti pod úroveň samotné fMLLR adaptace. V případě minimálního počtu adaptačních dat je však lépe nepoužít extrémní práh $Th_\gamma = 0,5$, kdy pro adaptaci zůstává akceptována pouze jedna ze složek stavu.

7.8.2 Inicializace lineárních transformací

Další z možností, jak robustně odhadnout neznámé parametry adaptace při omezeném množství dat, je inicializovat odhad matic lineárních transformací nějakou známou hodnotou. V podkapitole 6.2 byly popsány postupy inicializace adaptačních statistik statistik $\boldsymbol{k}_{(n)}$ a $\boldsymbol{G}_{(n)}$ (viz rovnice (3.39) a (3.38)) vhodnou hodnotou pro zvýšení robustnosti odhadu transformací $\boldsymbol{W}_{(n)}$ (z rovnice (3.33)). V podkapitole 6.2.1 byla popsána metoda, která interpoluje adaptační statistiky se statistikami získanými z SI modelu. Tyto statistiky však nepřidávají žádnou informaci o adaptovaném řečníku, pouze omezují odhad transformací směrem k SI modelu.

Další možností, jak zvýšit množství informace o řečníkovi pro adaptaci, je použít data od hlasově nejvíce podobných osob z trénovací databáze. Tyto tzv. N-best statistiky jsou využívané v metodách pro dotrénování SI modelu směrem k adaptačním datům (viz podkapitola 6.2.2).

Kombinace akumulovaných statistik

Akumulované statistiky od N-best řečníků lze přímo využít pro proces adaptace, ať už pro výpočet MLLR transformací [135], tak i pro jakoukoliv adaptaci založenou na akumulovaných statistikách (fMLLR, MAP nebo VTLN pomocí lineární transformace). V článku [136] jsme postup z [135] modifikovali pro rychlý odhad fMLLR transformačních matic.

Před samotnou adaptací jsou pro jednotlivé řečníky z trénovací databáze uloženy jejich nasčítané statistiky a natrénované GMM modely. Počet statistik pro jednoho řečníka odpovídá počtu složek ve všech stavech celého akustického modelu. Pro fMLLR jsou však jednotlivé statistiky shlukovány pomocí regresního stromu do omezeného počtu tříd (jejich počet odpovídá počtu koncových uzlů regresního stromu, viz podkapitola 3.4.4). Ukládáme proto pouze akumulované matice statistik $\boldsymbol{G}^s_{(n)}$, $\boldsymbol{k}^s_{(n)}$ řečníků s z trénovací databáze společně s jejich obsazením dané třídy daty $c^s_n = \sum_{b_{jm} \in C_n} \sum_t \gamma^s_{jm}(t)$, a to pro každou třídu C_n regresního stromu. Postup fMLLR adaptace s využitím naakumulovaných statistik od nejbližších řečníků je následující:

1. **Výběr kohorty N nejbližších řečníků:**
   Spočítáme logaritmus akustické věrohodnosti adaptačních dat neznámého řečníka oproti všem GMM modelům řečníků z trénovací databáze. Z těchto modelů vybereme N nejlepších podle velikosti vypočítané věrohodnosti. My však nepočítáme věrohodnosti celé adaptační promluvy najednou, ale použijeme plovoucí okénko s danou délkou a posunem. Pro vektory v aktuální pozici okénka vybereme nejlepší GMM. Okénkem posouváme po celé délce adaptační promluvy, tím dostaneme N nejlepších GMM modelů řečníků, kdy N je závislé na délce promluvy.

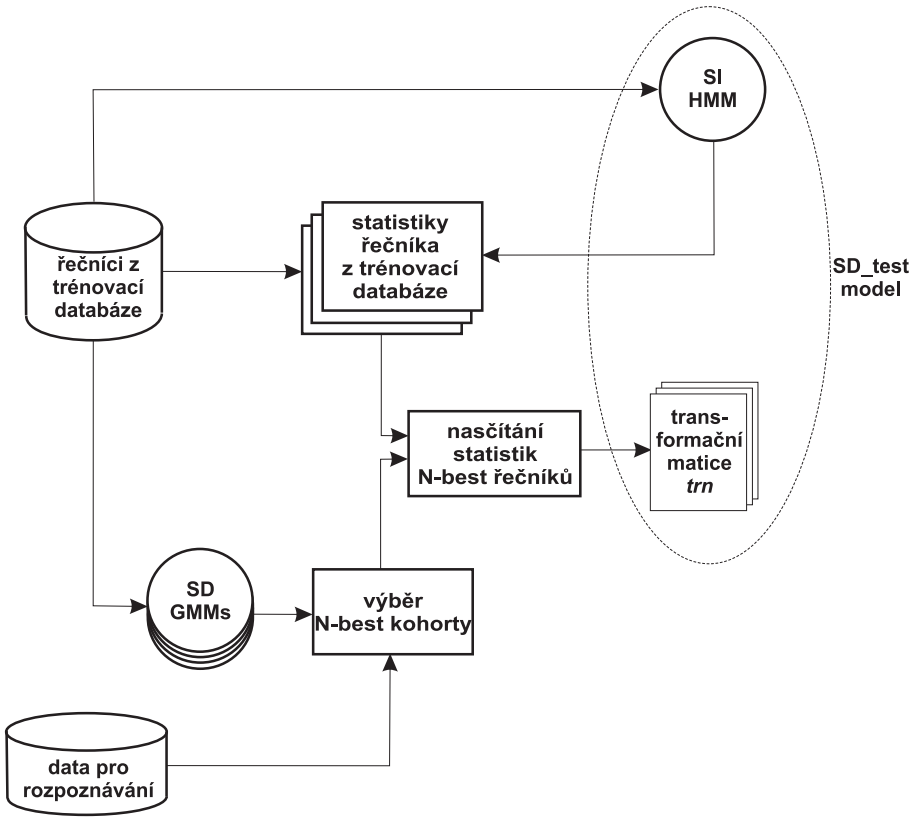
   Mezi GMM modely řečníků je přidán i **model univerzálního řečníka** (UBM – Universal Background Model) [89], který je konstruován stejně jako na řečníkovy nezávislý model (SI) v úloze rozpoznávání řeči (zde jde však o GMM). Ten se ale do kohorty nejbližších nepřidává, slouží pouze k odstranění neinformativních segmentů promluvy (např. neřečové události a pod.).

2. **Odhad fMLLR transformací:**
   Ve výpočtu fMLLR adaptace (podkapitola 3.4.2) jsou k statistikám aktuálního adaptovaného řečníka přidány také inicializační statistiky, tedy statistiky všech vybraných nejbližších řečníků $s = 1...N$, tzn.:

$$\boldsymbol{k}_{(n)i} = \sum_{s=1}^{N} \boldsymbol{k}^s_{(n)i} + \boldsymbol{k}_{(n)i} \ , \qquad \boldsymbol{G}_{(n)i} = \sum_{s=1}^{N} \boldsymbol{G}^s_{(n)i} + \boldsymbol{G}_{(n)i} \ , \tag{7.11}$$

pro každou n-tou regresní třídu C_n a i-tou řádku transformační matice $\boldsymbol{W}_{(n)}$, která je odvozena ML kritériem rovnicí (3.42).

Obrázek 7.9 ukazuje blokové schéma adaptace s vyžitím statistik od nejbližších řečníků. S rostoucím množstvím adaptačních dat se akumulují statistiky od většího množství řečníků, tím adaptovaný model pomalu konverguje k SI modelu. Tento proces tlumí vliv vlastních statistik adaptovaného řečníka. Pro určité kritické množství adaptačních dat již stačí samotné statistiky rozpoznávaného řečníka k dobrému odhadu transformačních matic fMLLR, není už potřeba k nim přidávat statistiky od nejbližších řečníků.



Obrázek 7.9: Blokové schéma kombinace statistik N-best řečníků pro adaptaci modelu s malým souborem adaptačních dat.

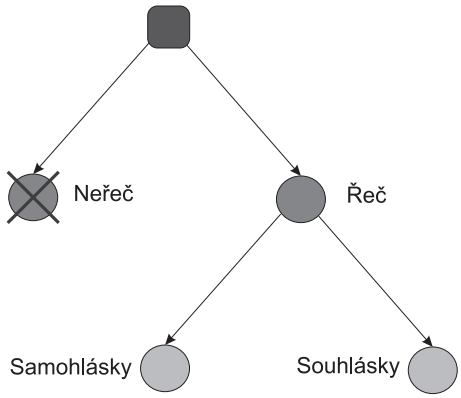
Další naší modifikací metody je možnost rozdělit adaptační data do fonetických kategorií (např. samohlásky/souhlásky) a hledat nejbližší řečníky a jejich statistiky s ohledem na danou fonetickou kategorii. Tento postup umožňuje větší variabilitu pro složení hlasu (statistik) rozpoznávaného řečníka z konečné množiny řečníků v trénovací množině.

Kombinace akumulovaných statistik s využitím fonetické informace

Možností, jak vylepšit předchozí výběr statistik, je zaměřit se na jejich vnitřní variabilitu. Neočekáváme, že jeden řečník z trénovací databáze bude mít stejný hlas jako adaptovaný řečník, když navíc máme k dispozici pouze omezenou trénovací databázi. Spíše než celková promluva

adaptovaného řečníka bude stejný způsob vyslovování některých částí jeho promluvy, např. některých fonémů, s výslovností stejných fonémů jiného řečníka. Jiné fonémy bude adaptovaný řečník vyslovovat obdobně jako další řečník. Nabízí se tedy možnost hledat inicializační statistiky od nejbližšího řečníka ne k celé rozpoznávané promluvě, ale rozdělit inicializační statistiky na menší úseky (např. na fonémy) a hledat k rozpoznávaným vysloveným fonémům jejich nejbližší podobné ze všech příslušných fonetických kandidátů na inicializaci. Vybrané inicializační statistiky pak nebudou pouze od nejbližších N řečníků, ale tyto jednotlivé statistiky budou inicializovat adaptaci s respektováním fonetické informace. Pro tuto inicializaci je nutné nejprve všechna potenciální data od řečníků z trénovací databáze rozdělit podle jejich fonetické informace.

Při adaptaci, která je uvažována v této části práce, je dostupné velmi malé množství adaptačních dat, a tedy ne všechny fonémy jsou z tohoto předpokladu pozorovatelné v adaptačních datech. Přesto bychom chtěli mít v inicializačních datech i fonémy, které nebyly v adaptační promluvě obsažené. Možností je tedy namísto shlukování parametrů modelu na základě jejich blízkosti v akustickém prostoru použít fonetickou informaci, tedy shlukovat parametry modelu podle toho, jaký foném reprezentují (v případě trifónů jde o foném definovaný prostředním stavem). Je možno místo klasického regresního stromu použít regresní strom v závislosti na fonetických vlastnostech. Více o shlukování blízkých parametrů modelu viz podkapitola 3.4.4. Pro naši úlohu nám však vystačí mnohem menší regresní strom, než je uvedeno v podkapitole 3.4.4, zde si vystačíme pouze se třemi fonetickými třídami – samohlásky, souhlásky a neřečové události (viz obrázek 7.10).



Obrázek 7.10: Fonetický strom pro inicializaci statistik s využitím fonetické informace.

Modifikujeme postup inicializace statistik s využitím fonetické znalosti [137]:

- **Akumulace statistik z trénovací databáze** – pro každého řečníka s z trénovací databáze jsou naakumulované matice statistik $\boldsymbol{k}^s_{(n)}$ and $\boldsymbol{G}^s_{(n)}$ (viz (3.39) a (3.38)) jen s tím rozdílem, že třídy $C_n, n = 1, \ldots, N$ jsou dány fonetickým regresním stromem. Je tedy nutné získat fonetický přepis trénovacích dat, aby je dále šlo rozdělit na dané třídy a nad každou takovou třídou pro jednotlivé řečníky natrénovat GMM.

- **Výběr nejbližších statistik** – data od adaptovaného řečníka jsou rozdělena do tříd fonetického regresního stromu díky fonetické informaci v jejich přepise. Pro tato data v každé třídě jsou nalezena nejbližší podobná data (statistiky) od řečníků z trénovací databáze. Pro třídu nedostatečně obsazenou rozpoznávanými daty je uvažována nadřazená

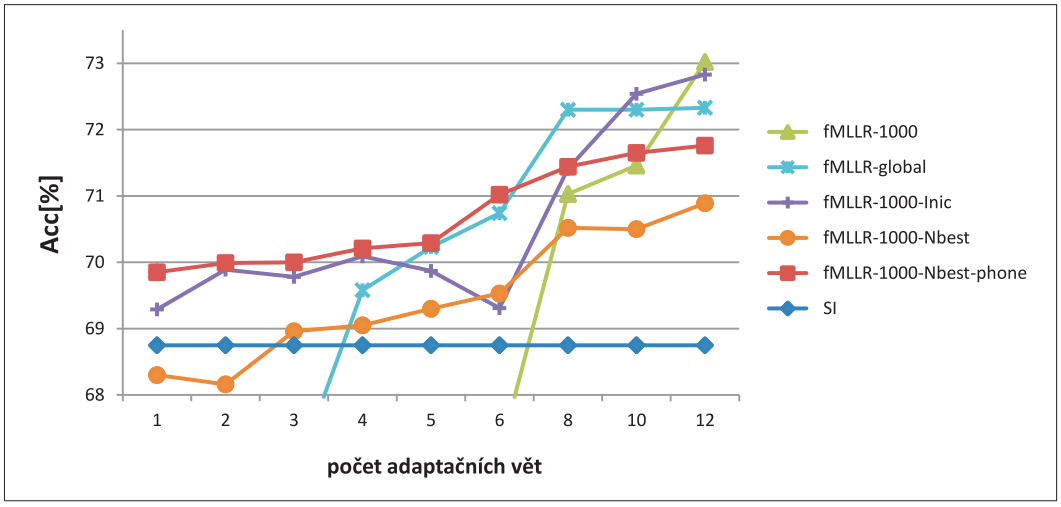
třída ve fonetickém regresním stromě. K nalezení nejbližších statistik jsou použity GMM modely a je zvolen stejný postup jako v předchozí metodě, tedy maximalizujeme akustickou věrohodnost adaptačních dat v plovoucím okénku oproti všem GMM modelům a vybíráme tak nejpodobnější statistiky přiřazené nejlepším GMM.

- **Nasčítání podobných statistik** – matice akumulovaných statistik (3.39) a (3.38) jsou inicializovány vybranými daty z druhého kroku. Nakonec jsou k těmto inicializačním statistikám přidána i aktuální data rozpoznávaného řečníka.

Hlas adaptovaného řečníka je nyní reprezentován ne jen průměrným hlasem jeho nejbližších napodobitelů z trénovací databáze, ale je využita i fonetická informace a inicializační statistiky jsou tedy po částech složeny z průměrných fonetických událostí v hlase adaptovaného řečníka.

Výsledky inicializačních metod

V grafu 7.11 a v tabulce B.4 v Přílohách lze nalézt výsledky různých inicializací metody fMLLR (s regresním stromem s $Th = 1000$), a to jmenovitě inicializace statistikami z SI modelu z podkapitoly 6.2.1 (označeno jako fMLLR-1000-inic, kde množství dat je určeno váhou jednotlivých složek modelu SI), inicializace N nejbližšími řečníky z trénovací databáze (označeno jako fMLLR-1000-Nbest) a inicializace N nejbližšími řečníky s využitím fonetické informace (označeno jako fMLLR-1000-Nbest-phone). Množství inicializačních dat u metod založených na N nejbližších řečnících je dáno velikostí plovoucího okénka (30 vzorků s posunem 10 vzorů), pro výběr kohorty je použita pouze první z adaptačních vět. V grafu jsou též zaneseny výsledky pro samotnou metodu fMLLR (s respektováním regresního stromu i s globální maticí, označeny jako fMLLR-1000 a fMLLR-global) a výsledky neadaptovaného SI modelu (označení SI).



Obrázek 7.11: Výsledky (Acc[%]) inicializace metody fMLLR s různou volbou inicializace statistik pro korpus SD-E. Pro porovnání uvedeny i výsledky samotné adaptace fMLLR (globální i s regresním stromem s $Th = 1000$) a výsledky neadaptovaného SI modelu.

Výsledky uvedené v grafu 7.11 opodstatňují inicializaci metody fMLLR, která pro nízké množství adaptačních dat významně zhoršuje adaptaci. Při inicializaci je sice vliv adaptace

utlumen při dostatečném množství adaptačních dat, ale zato je kompenzována chyba způsobená samotnou fMLLR pro malé množství dat. Nejlépe vychází metoda fMLLR s navrženou inicializací statistikami od N nejbližších řečníků z trénovací databáze s respektováním fonetické informace (fMLLR-1000-Nbest-phone). Tento přístup překonává inicializaci modelem SI (fMLLR-1000-inic), protože k inicializaci využívá statistiky bližší fonémům adaptovaného řečníka. Naopak inicializace bez fonetické informace (fMLLR-1000-Nbest) za zmíněnými metodami zaostává. Zdůvodnění lze nalézt právě v lokální rozdílnosti inicializačních statitik (např. na úrovni fonémů), i když globálně jde o podobná data.

7.8.3 Adaptace založená na kombinaci bázových matic

Dalším postupem, jak snížit počet odhadovaných parametrů pro adaptaci, je reprezentovat transformační matice v nižším podprostoru definovaném pomocí bázových matic. Hledaná transformační matice $\boldsymbol{W}$ adaptovaného řečníka je dána lineární kombinací bázových vektorů. Bázové matice jsou určovány z trénovacích dat před započtením adaptačního procesu, tedy bez znalosti dat adaptovaného řečníka. Pomocí adaptačních dat jsou hledány pouze váhové koeficienty lineární kombinace, tedy podstatně menší počet neznámých než při odhadu celé transformační matice metodou (f)MLLR.

Popis této metody spolu s bázovými maticemi odvozenými pomocí EV a ML odhadu lze nalézt v podkapitole 6.6. Níže uvedeny jsou další možné postupy pro volbu bázových matic, které jsme uvedly a zhodnotili v článku [138].

Transformační matice trénovacích řečníků

Naivní přístup k nalezení bázových matic je využít přímo transformační matice od velkého množství řečníků z trénovací databáze. Problém je, jak z takového množství matic vybrat ty nejvíce informativní. Možností je řečníky v trénovací databázi shlukovat a použít pouze transformační matice natrénované na všech datech daného shluku. Transformační matice je možno si vypočítat off-line pro různý počet shluků (pro různou velikost počtu kombinovaných bází B).

Báze definovaná faktorovou analýzou

Jak bylo zmíněno v podkapitole 6.5, faktorová analýza je statistickou alternativou k dekompozici vlastních vektorů (EV viz podkapitola 6.6.1). Bázové matice $\boldsymbol{W}_b$ z (6.28) jsou zde reprezentovány sloupci matice faktorových zátěží $\boldsymbol{L}$. Pro odhad faktorových zátěží byl použit iterativní algoritmus založený na ML. Důležité je si uvědomit, že potřebujeme vždy jiný počet faktorů (počet kombinovaných bází B je dán množstvím adaptačních dat), ale u předem vypočítaných faktorů nelze (jako v EV) určit jejich významnost. Je proto nutné off-line vypočítat různé matice $\boldsymbol{L}$ pro různý počet B a z těchto matic se pak při vlastní adaptaci vybere ta, která odpovídá aktuálnímu počtu adaptačních dat.

Analýza nezávislých komponent (ICA)

Alternativní postup k nalezení vhodné bázové reprezentace podprostoru je založen na **analýze nezávislých komponent** (ICA – Independent Component Analysis) [139]. Jde o metodu

hojně využívanou k separaci zdrojových signálů. Předpokládáme lineární ICA model, kde pozorování $\boldsymbol{o}(t) = (o_1(t), \ldots, o_D(t))$ je rozloženo na komponenty $\boldsymbol{s} = (s_0, \ldots, s_{K-1})$ pomocí lineární statistické transformace $\boldsymbol{A}$

$$\boldsymbol{o}(t) = \boldsymbol{A}\boldsymbol{s}. \tag{7.12}$$

Máme-li tento model a testovací data $\boldsymbol{Z} = (\boldsymbol{o}(1), \ldots, \boldsymbol{o}(T))$, úkolem je nalézt mixážní matici $\boldsymbol{A}$ a zdroj $\boldsymbol{s}$. Inverzní matice $\boldsymbol{A}^{-1}$ se nazývá separační maticí, $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T]$ je maticí nezávislých komponent

$$\boldsymbol{S} = \boldsymbol{A}^{-1}\boldsymbol{Z}. \tag{7.13}$$

Separace signálů v pozorovaných datech je prováděna tak, aby výsledné komponenty $\boldsymbol{s_i}$ byly nezávislé a jejich rozdělení negausovské. Gausovská vlastnost, která je předpokládána v metodách (EV a FA), nedovoluje objevit rotaci v latentním prostoru (prostoru nezávislých komponent) [140]. ICA přístup je tedy méně omezující pro hledání komponent vstupního signálu.

Maximalizujeme funkci měřící nezávislost komponent. Při odhadu nezávislosti se využívá **centrální limitní věta** (CLT – Central Limit Theorem), součet jakýchkoli *iid* náhodných proměnných se blíží k normálnímu rozdělení. Její užití je však v opačném směru, snahou je tedy nalézt takové komponenty, které se co nejvíce liší od normálního rozdělení.

Algoritmus ICA pracuje se signály obsahujícími střední hodnotu, avšak operace se signály bez střední hodnoty jsou jednodušší, proto je obvyklé data nejprve centrovat.

Jako bázové matice pro vztah (6.28) volíme vektory matice $\boldsymbol{A}$. Stejně jako v FA nelze vliv jednotlivých ICA vektorů posuzovat podle některé dodatečné informace (jako je vlastní číslo v EV), proto je nutno určit počet bázových matic B off-line, tedy již při výpočtu ICA. Prakticky je off-line vypočteno více matic $\boldsymbol{A}$ pro různý počet B a z těchto matic se pak při vlastní adaptaci vybere ta, která je pro aktuální počet rozpoznávaných dat nejlepší.

Výsledky pro různou volbu bázových matic

Výsledky testů pro různé volby bázových matic pro odhad globální matice fMLLR lze nalézt v grafu 7.12 a v tabulce B.5 v Přílohách, kde označení bází je následující: Wnode – je báze daná maticemi shluků trénovacích řečníků, FA – báze daná faktorovou analýzou, ICA – báze určená z analýzy nezávislých komponent[2]. Jde o vlastní postupy uvedené výše v této podkapitole. Dále je v grafu uvedeno ML – odhad bázových matic vycházející z ML kritéria a EV – báze definovaná největšími vlastními vektory (odvozeny metodou SVD). Jde o postupy popsané v podkapitole 6.6. Pro porovnání jsou uvedeny výsledky fMLLR globální adaptace a výsledky s neadaptovaným SI modelem.

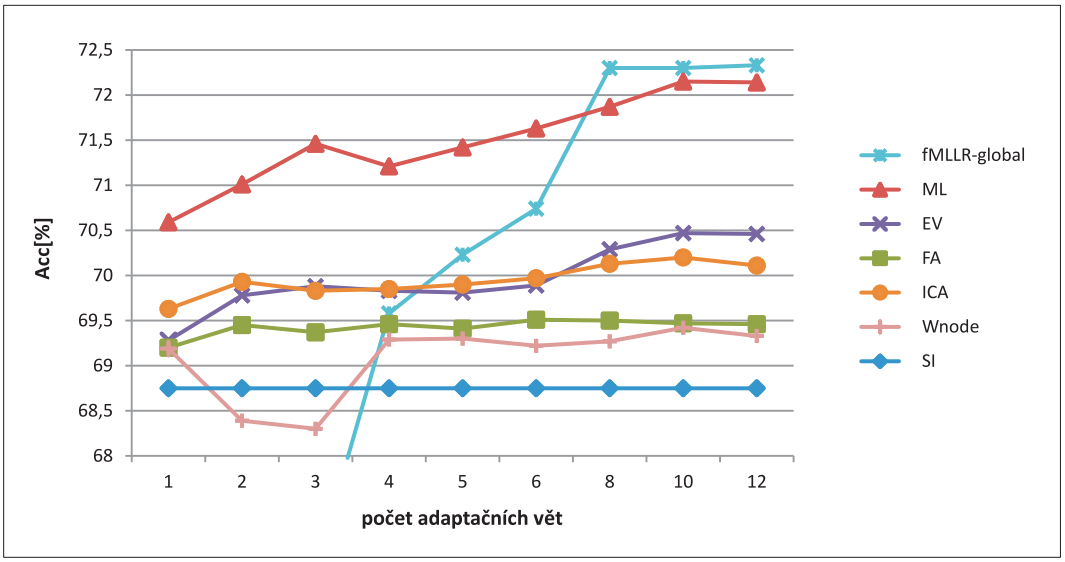
Množství bázových matic B pro odhad adaptace bylo voleno dynamicky v závislosti na velikosti adaptační množiny [112]

$$B = \min(\eta\beta, d(d+1)), \tag{7.14}$$

kde η je apriori volená konstanta (v této práci $\eta = 0,2$), β je množství akumulovaných statistik náležící adaptovanému řečníku (viz rovnice (3.44)) a d je dimenze akustického vektoru, $d(d+1)$ je dimenze hledané transformační matice $\boldsymbol{W}$[3].

[2] pro výpočet ICA jsme využili program `http://www.cis.hut.fi/projects/ica/fastica/`

[3] některé metody ze své podstaty dokáží najít maximálně T bázových matic, kde T je počet trénovacích dat



Obrázek 7.12: Výsledky (Acc[%]) adaptace fMLLR s různou volbou bázových matic, pro SD-E korpus. Pro porovnání uvedeny i výsledky globální adaptace fMLLR a výsledky neadaptovaného SI modelu.

Ze všech uvedených postupů jednoznačně nejlépe vychází metoda založená na ML odhadu bázových matic navržená v práci 6.6.1, i když i ostatní přístupy kompenzují chyby fMLLR adaptace způsobené malým množstvím adaptačních dat.

7.8.4 Redukce informace pomocí neuronové sítě

Naše idea je založena na redukci chybné informace ze špatně odhadnuté adaptace, tedy adaptace na malém množství adaptačních dat [141]. ANN (viz podkapitola 6.7) je natrénována na trénovacích párech [chybně odhadnutá adaptace; korektně odhadnutá adaptace]. Neuronová síť pak redukuje vliv špatně odhadnutých parametrů adaptace, ale ponechá informaci od parametrů, které byly odhadnuty správně. Korektně odhadnutá adaptace je získána odhadem s dostatečným množstvím adaptačních dat.

Možné využití ANN, konkrétně pak sítě bottleneck (popsané v podkapitole 6.7.2), je při redukci informace v adaptační matici $\boldsymbol{W} = [\boldsymbol{A}, \boldsymbol{b}]$. Vstupem/výstupem ANN je supervektor $\boldsymbol{w} = \text{vec}(\boldsymbol{W})$ zformovaný z řádků matice $\boldsymbol{W}$ pospojovaných za sebou do vysoko-dimenzionálního vektoru – supervektoru. Limitací tohoto přístupu je však právě formát vstupních/výstupních dat.

Poznamenejme, že redukovat pomocí ANN lze například i supervektor všech středních hodnot akustického modelu. Komplikací tohoto přístupu je obrovská velikost dimenze takového supervektoru, pro natrénování ANN by bylo zapotřebí velkého množství trénovacích párů. Tato práce se však orientuje z velké části na adaptaci založenou na lineárních transformacích, proto je aplikace ANN směřována spíše tímto směrem.

Problém sítě bottleneck pro fMLLR

Matice $\boldsymbol{W}$ musí být transformována pro účely ANN do tvaru supervektoru $\boldsymbol{w}$. S $\boldsymbol{w}$ je uvnitř ANN zacházeno jako s vektorem (tedy veškerá informace o původním maticovém uspořádání je ztracena) a teprve výstupní vektor $\boldsymbol{w_{out}}$ je opět zformován do matice $\boldsymbol{W_{out}}$. Vlastnosti lineárního prostoru popisovaného původní maticí $\boldsymbol{W}$ jsou tímto procesem značně porušeny a tedy výstupní matice $\boldsymbol{W_{out}}$ popisuje naprosto odlišný prostor, což není naším cílem. My chceme pouze redukovat informaci od špatně odhadnutých parametrů adaptace.

Podobný problém řeší i přístup využívající bázových vektorů popsaný v podkapitole 6.6. Pro nalezení bázových vektorů je zde matice $\boldsymbol{W}$ také transformována do tvaru vektoru $\boldsymbol{w}$, avšak finální matice $\boldsymbol{W_{out}}$ je vybrána s ohledem na maximalizaci věrohodnosti adaptačních dat.

Z důvodu vyhnutí se tomuto problému byla pro redukci informace využita metoda shiftMLLR popsaná v podkapitole 6.1, která odvozuje pro adaptaci řečníka pouze matici posuvu $\boldsymbol{b}$ a ignoruje matici $\boldsymbol{A}$. U této metody již z jejího principu odpadá nutnost transformace matice $\boldsymbol{W} = [\boldsymbol{b}]$ do tvaru vektoru.

Redukce shiftMLLR pomocí ANN bottleneck

Navržený postup pro redukci dimenze transformace shiftMLLR má následující strukturu [141]:

- **Formát dat**: $\boldsymbol{w}_s = [\boldsymbol{b}^{\mathrm{T}}_{s(1)}, \ldots, \boldsymbol{b}^{\mathrm{T}}_{s(N)}]^{\mathrm{T}}$ je vstupní vektor s-tého řečníka – v případě využití více transformací pro jednoho řečníka, kde N udává počet transformací. Všechny transformační vektory $\boldsymbol{b}_{s(n)}, n = 1, \ldots, N$ jsou pospojovány do jediného supervektoru. Počet transformací N musí být stejný pro každého řečníka. Dimenze supervektoru je $D = N \cdot d$, kde d je dimenze vektoru pozorování, a tedy i dimenze jedné transformace.

- **Trénování**: Vstupní supervektory $\boldsymbol{w}_s^{\text{train}}$ jsou odvozeny pro každého řečníka z trénovací sady pomocí shiftMLLR adaptace pouze z malého množství adaptačních dat. Výstupní vektory $\boldsymbol{w}_s^{train-out}$ (informace od učitele – supervised trénování) jsou poskládány z transformací shiftMLLR odvozených ze všech dostupných dat od řečníků z trénovací databáze. Neuronová síť je natrénována na trénovacích párech $[\boldsymbol{w}_s^{\text{train}}; \boldsymbol{w}_s^{train-out}], s = 1, \ldots, S$. Je tedy natrénována nelineární transformace vstupu na požadovaný výstup, ANN má naučené relace mezi špatně a dobře podmíněnými adaptačními transformacemi shiftMLLR. Natrénovaná síť bottleneck by měla odstraňovat nekonzistenci mezi zadaným vstupem a výstupem.

- **Testování**: Poté, co byl akustický model adaptován metodou shiftMLLR, je zkonstruován supervektor $\boldsymbol{w}^{\text{test}}$ a propagován skrz natrénovanou síť bottleneck pro získání výstupního supervektoru $\boldsymbol{w}^{\text{test-out}}$. Tento výstupní supervektor $\boldsymbol{w}^{\text{test-out}} = [\boldsymbol{b}^{\text{test-out}}_{(1)}, \ldots, \boldsymbol{b}^{\text{test-out}}_{(N)}]$ (s redukovanou informací) je transformován zpět do tvaru transformace shiftMLLR a použit pro adaptace původního akustického modelu.

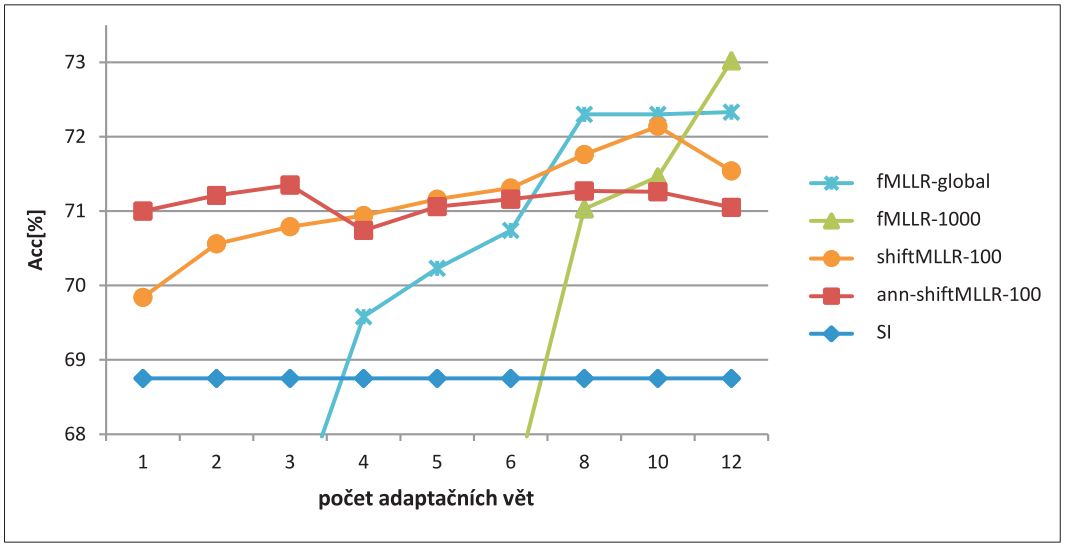
Výsledky redukce informace pomocí ANN

Pro nastavení tohoto testu jsme použili regresní strom s 64 listovými uzly, tedy vstupní vektor pro ANN je složen z $N = 64$ transformačních vektorů $\boldsymbol{b_{(n)}}$. Práh okupace regresního stromu byl nastaven na $Th = 100$. Třívrstvá síť bottleneck byla natrénována metodou IRPROP (viz podkapitola 6.7.1). Pro účely shiftMLLR adaptace s 64 transformacemi byl počet

neuronů v jednotlivých vrstvách 2112, 100, 2112. Topologie sítě je zobrazena na obrázku 6.5, kde $D = 2112$ a $B = 100$. Ve skrytých vrstvách byla použita sigmoidální aktivační funkce a ve výstupní vrstvě pak lineární aktivační funkce.

ANN byla natrénována na 700 řečnících z trénovací databáze SD-E korpusu. Vstupní vektory shiftMLLR adaptace byly odvozeny pouze pro 1 a 2 adaptační věty, tedy bylo použito cca 20 vstupních vektorů natrénovaných na různých větách pro každého řečníka. Každý výstupní vektor byl vytvořen s využitím všech dostupných 50 vět od každého řečníka. Poznámka: všech 20 vstupních vektorů od daného řečníka má přiřazen stejný výstupní vektor od tohoto řečníka. Úkolem ANN je najít relaci mezi špatně odhadnutými adaptacemi (vstup) a těmi dobře odhadnutými (výstup).

V testovací fázi je aktuální shiftMLLR adaptace propagována natrénovanou sítí a výstupní adaptací (adaptací s redukovanou chybovou informací) je pak adaptován akustická model. Výsledky rozpoznávání s takto adaptovaným modelem jsou zobrazeny v grafu 7.13 a v tabulce B.6 v Přílohách.

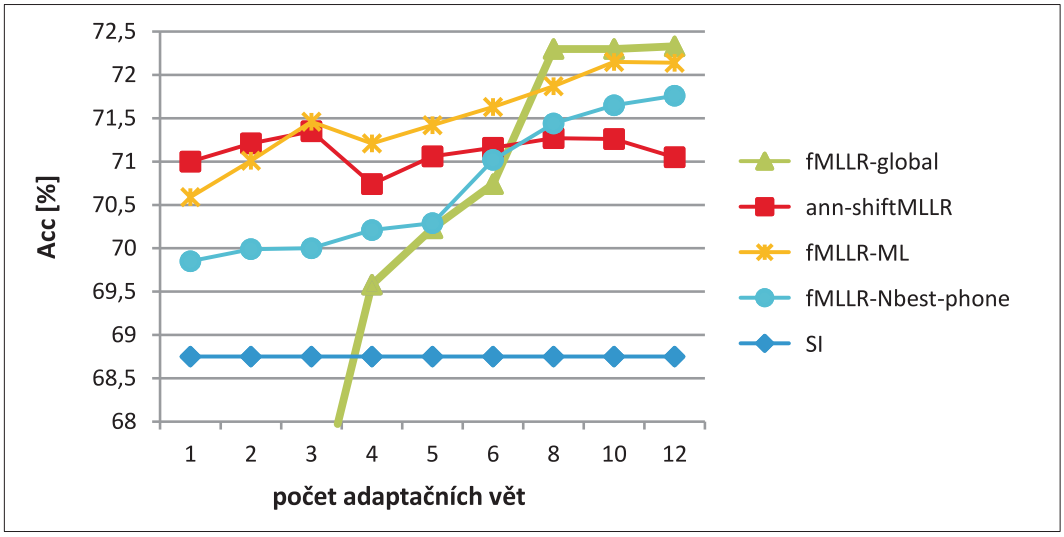


Obrázek 7.13: Výsledky (Acc[%]) adaptace shiftMLLR s a bez využitím ANN pro zrobustnění adaptace, pro SD-E korpus. Pro porovnání uvedeny i výsledky adaptace fMLLR (globální i s regresním stromem s $Th = 1000$) a výsledky neadaptovaného SI modelu.

Přístup zrobustněné adaptace shiftMLLR pomocí ANN přináší znatelné zlepšení adaptační metody použité pro malé množství adaptačních dat, důvodem je natrénování sítě pro případ adaptace pouze s jednou a dvěma adaptačními větami. Síť bottleneck dle předpokladů odstraní nežádoucí informaci z adaptace, tedy chybnou informaci, která zhoršuje výsledek rozpoznávání.

7.9 Porovnání nejlepších adaptačních přístupů

V této podkapitole jsou porovnány navržené adaptační metody s nejlepšími výsledky pro malý počet adaptačních dat popsaných v této práci. Na souhrnné výsledky lze nahlédnout v grafu 7.14. Porovnávané adaptační metody jsou fMLLR s jednou globální transformací odha-



Obrázek 7.14: Výsledky (Acc[%]) adaptace fMLLR s ML odhadem bázových matic (fMLLR-ML), adaptace fMLLR s inicializací od N nejbližších řečníků s využitím fonetické informace (fMLLR-Nbest-phone) a adaptace shiftMLLR s využitím ANN pro zrobustnění adaptace (ann-shiftMLLR), pro korpus SD-E. Pro porovnání uvedeny i výsledky globální adaptace fMLLR (fMLLR-global) a výsledky neadaptovaného SI modelu.

dovanou jako lineární kombinace bázových matic ML odhadem z podkapitoly 6.6.1 (v grafu značena fMLLR-ML), dále pak v této práci v podkapitole 7.8.2 navržená metoda inicializace fMLLR (s regresním stromem s $Th = 1000$) od N nejbližších řečníků s využitím fonetické informace (s označením fMLLR-Nbest-phone) a metoda shiftMLLR s regresním stromem (s regresním stromem s $Th = 100$), ale s pevným počtem použitých transformací $N_{trn} = 64$ zrobustněná průchodem přes ANN bottleneck navrženou v podkapitole 7.8.4 (označena ann-shiftMLLR). Pro porovnání jsou v grafu uvedeny i výsledky klasické metody fMLLR pouze s globální transformací (fMLLR-global) a výsledky neadaptovaného SI modelu.

Z porovnání těchto přístupů vychází nejlépe metody fMLLR-ML a ann-shiftMLLR, které prokazují srovnatelnou úspěšnost pro malý počet adaptačních dat. Při extrémně nízké adaptační množině (1 adaptační věta, v SD-E korpusu odpovídá cca 4 sekundám pro unsupervised adaptaci) vykazuje ann-shiftMLLR mírný náskok nad fMLLR-ML až o 0,41 % Acc absolutně, na hladině významnosti 95 %. Metoda fMLLR-ML pro větší počet adaptačních dat (6 – 12 vět) dokazuje rostoucí úspěšnost, kde ann-shiftMLLR stagnuje. To je z velké části způsobeno natrénováním ANN pouze pro případy adaptace s jednou a dvěma adaptačními větami. To byl však náš požadavek, natrénovat ANN pro tyto případy.

7.10 Zhodnocení experimentů

Experimenty provedené na českém telefonním korpusu CzT s dostatečným množstvím dat pro adaptaci a velkým počtem různých řečníků dokázaly opodstatnění adaptace na řečníka. Z výsledků jsou také patrné výhody a nevýhody jednotlivých metod, jejich rychlost a účinnost. Například metoda MAP se ukázala být dobrou volbou pro první iterační krok. Pomocí ní se

změní adaptačními daty dobře podmíněné složky modelu, ostatní složky jsou pak v druhé iteraci zpřesněny jinou metodou, např. fMLLR s výhodou adaptace vektoru pozorování. Model předpřipravený pomocí adaptačního trénování (SAT, VTLN) dokázal zvýšit účinnost jednotlivých metod v porovnání s klasicky natrénovaným modelem.

Nevýhodou klasických metod adaptace představených v kapitole 3 je jejich slabá účinnost v úloze s malým počtem adaptačních dat. To bylo dokázáno testy provedenými na korpusu SD-E, kde i metody založené na shlukování podobných příznaků akustického modelu (metody (f)MLLR) zhoršovaly rozpoznávání pro malý počet adaptačních vět. Proto byly v kapitole 6 představeny robustní přístupy k adaptaci mající za cíl eliminovat zhoršení rozpoznávání díky špatně odvozené adaptaci zapříčiněné nedostatkem adaptačních dat. Tyto metody pak byly otestovány spolu s vlastními návrhy na zvýšení robustnosti adaptace v podkapitole 7.8.

Byly navrženy tři vlastní přístupy (a jeden modifikován) ke zvýšení robustnosti adaptačních metod založených na lineárních transformacích. První z nich eliminuje nepřesně zarovnaná dat pro adaptaci (podkapitola 7.8.1). Druhý inicializuje chybějící adaptační data pro odhad transformace daty od nejpodobnějších řečníků z trénovací databáze (podkapitola 7.8.2). Třetím přístupem je využití neuronové sítě pro zvýšení robustnosti adaptace s malým počtem adaptačních dat. Poslední přístup minimalizuje počet odhadovaných neznámých proměnných pouze na počet vah lineární kombinace bázových matic a spočívá ve vhodném odhadu bázových matic (vlastní modifikací pak volba těchto bázových matic).

Všechny tyto metody dokázaly své opodstatnění v úloze adaptace s malým počtem dat, kde odstranily chyby způsobené klasickými metodami adaptace. Z výsledků testování pak vyplynuly dvě metody s porovnatelnými výsledky: metoda lineární kombinace bázových matic získaných pomocí ML odhadu, navržená v práci [112], a v této práci navržená metoda pro zvýšení robustnosti adaptace shiftMLLR pomocí ANN bottleneck. Obě tyto metody jsou pro malý počet adaptačních dat srovnatelné.

Kapitola 8

Závěr

Problém adaptace akustického modelu v úloze rozpoznávání spojité řeči je již dlouhou dobu řešen množstvím vědeckých pracovišť po celém světě. Existuje velké množství metod a přístupů v různých oblastech zpracování jak modelu tak i signálu. Přesto jde stále o otevřený problém. Jak dochází k zrychlování výpočtů a tím k zpřesňování samotného akustického modelu, objevují se nové přístupy k adaptaci, které vykazují větší účinnost nebo naopak rychlost adaptace pro použití v reálném čase, kdy je akustický model adaptován za běhu řečového rozpoznávače. Tyto dva problémy (rychlost a přesnost) jsou si navzájem v protikladu.

Cílem této práce bylo prostudovat stávající přístupy k adaptaci akustického modelu v úloze rozpoznávání spojité řeči a to jak generativní, tak i diskriminativní metody a nalézt jejich silné a slabé stránky. Tyto metody jsou popsány v kapitole 3. Pro ucelený pohled na adaptaci bylo potřebné také zmínit adaptační přístupy pro trénování, které se aplikují na trénovací data, z kterých je pak vytvořen akustický model bez rušivé informace o řečníkovi. Tyto metody jsou popsány v kapitole 4. Experimentální testování metod proběhlo na dvou rozdílných datových korpusech. Výsledky společně s komentáři k vzájemnému porovnání těchto metod jsou uvedeny v podkapitolách 7.4,7.5 a 7.7. Ukázalo se, že diskriminativní přístupy k adaptaci vyžadují k dobrému natrénování podstatně větší množství dat než generativní přístupy. Metoda MAP se ukázala pro úlohu s malým počtem adaptačních dat nevhodnou. Naopak metody založené na lineárních transformacích (LT) jsou pro tento problém přímo navrženy, avšak při extrémně malém počtu adaptačních dat přesto selhávají (dochází ke špatnému odhadu velkého množství parametrů transformací a tím i ke zhoršení rozpoznávání).

Dále bylo cílem práce definovat problémy provázející on-line adaptaci, tedy problémy úzce související s on-line zpracováním mluvené řeči. Při on-line rozpoznávání není známa identita řečníka ani referenční přepis žádné části jeho dat, proto adaptace musí proběhnout až v průběhu rozpoznávacího procesu na aktuálně rozpoznávaných datech. Hlavním problémem on-line adaptace je obvykle malý tok adaptačních dat kontrastující s požadavkem rychlé adaptace na řečníka. Tyto problémy byly rozepsány v kapitole 5. S úspěchem byla do online systému implementována metoda fMLLR s drobnými úpravami (transformace neřečových událostí, kontextové CF), výsledky testování spolu s popsáním experimentu uvedeny v podkapitole 7.6.

Z konkrétních problémů on-line adaptace byla práce nejvíce zaměřena na zvyšování robustnosti adaptace systému při využití velmi malého množství adaptačních dat. Tomuto problému, a možných přístupů k jeho vyřešení používaných ve světě, byla věnována kapitola 6. V podkapitole 7.8 byly pak společně s výsledky testování těchto robustních metod pro různé množství adaptačních dat uvedeny i vlastní přístupy a modifikace adaptačních přístupů pro dosažení

větší robustnosti pro malý počet dat. Tyto výsledky byly navzájem porovnány a okomentovány. Robustní přístupy pro adaptaci dokázali (některé více, jiné méně) odstranit problémy způsobené nízkým počtem dat od adaptovaného řečníka, tedy zabránit špatnému odhadu adaptace a přitom zachovat výhody, pro které je adaptace v ASR hojně využívána.

Konkrétně metody založené na lineárních transformací se ukázaly nejvhodnější pro problematickou úlohu adaptace s malým množstvím dat. Byly navrženy tři vlastní inovativní přístupy k robustní adaptaci. První z nich eliminuje nepřesně zarovnaná dat pro adaptaci. Druhý inicializuje chybějící adaptační data pro odhad transformace daty od nejpodobnějších řečníků z trénovací databáze. Třetím přístupem je využití neuronové sítě pro odstranění rušivé informace ze špatně odhadnuté adaptace. Tyto tři vlastní přístupy byly porovnány s momentálně nejlepší metodou pro robustní adaptaci, která minimalizuje počet odhadovaných neznámých proměnných pouze na počet vah lineární kombinace bázových matic získaných pomocí odhadu ML. Hlavní cíl zamezit špatnému odhadu transformačních matic při malém množství dostupných dat a tím se vyhnout možné degradaci přesnosti rozpoznávání splnily všechny navržené postupy.

8.1 Shrnutí přínosů práce

- Popsány, programově realizovány a otestovány
  - klasické i diskriminativní přístupy adaptace.
  - adaptační přístupy k trénování akustického modelu.
  - známé robustní přístupy k adaptaci s malým množstvím dat.
- Navrženy a experimentálně ověřeny
  - přístupy k rychlé kombinaci dvou klasických metod (fMLLR a MAP).
  - modifikace ohodnocení jistoty dat pro unsupervised adaptaci.
  - různé volby bázových matic pro jednu z robustních metod adaptace.
  - tři vlastní metody pro robustní adaptaci, které byly a porovnány s ostatními přístupy.
- Metody on-line adaptace implementovány do reálného ASR.

Stanovené cíle disertační práce byly splněny, dalším směrem k zlepšování robustní adaptace by měla být úprava předzpracování dat pro ANN k využití redukce chybné informace v plných transformacích fMLLR natrénovaných při malém počtu adaptačních dat.

Literatura

[1] F. Jurčíček, A. Pražák, L. Müller, J. Psutka, and L. Šmídl, "Design of LVCSR decoder for Czech language," in *ECMS*, Liberec, 2003, pp. 39–43.

[2] J. Rajnoha and P. Pollák, "ASR systems in noisy environment: Analysis and solutions for increasing noise robustness," *Radioengineering*, vol. 20, no. 1, pp. 74–84, 2011.

[3] A. Pražák, J. V. Psutka, J. Hoidekr, J. Kanis, L. Müller, and J. Psutka, "Automatic online subtitling of the Czech parliament meetings," *Lecture Notes in Artificial Intelligence*, vol. 4188, pp. 501–508, 2006.

[4] J. Psutka, L. Müller, J. Matoušek, and V. Radová, *Mluvíme s počítačem česky*. ACADEMIA Praha, 2006.

[5] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, 2001-2006.

[6] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[7] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[8] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[10] V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *IEEE Transactions On Speech and Audio Processing*, vol. 3, no. 3, pp. 357–366, 1995.

[11] K. Yu, "Adaptive training for large vocabulary continuous speech recognition," Ph.D. dissertation, Hughes Hall College and Cambridge University Engineering Department, 2006.

[12] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2003.

[13] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition." *Computer Speech and Language*, vol. 16, pp. 25 – 47, 2002.

[14] Y. Chow, "Maximum mutual information estimation of HMM parameters for continuous speech recognition using the N-best algorithm," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Albuquerque, 1990, pp. 701–704.

[15] D. Povey and P. Woodland, "Frame discrimination training of HMMs for large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, 1999, pp. 333–336.

[16] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Eurospeech*, Lisbon, 2005, pp. 2133–2136.

[17] R. Schlüter and W. Macherey, "Comparison of discriminative training criteria," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Seattle, 1998, pp. 493–496.

[18] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," in *Interspeech*, Lisboa, 2005, pp. 2125–2128.

[19] F. Wessel, R. Schltiter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 288–298, 2001.

[20] J.-L. Gauvain and C.-H. Lee, "Maximum a-posteriori estimation for multivariate gaussian mixture observations of Markov chains," *IEEE Transactions On Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[21] Y. Gao, B. Ramabhadran, and M. Picheny, "New adaptation techniques for large vocabulary continuous speech recognition," in *ICSA ITRW ASR*, Paris, 2000, pp. 107–111.

[22] D. Povey, M. Gales, D. Kim, and P. Woodland, "MMI-MAP and MPE-MAP for acoustic model adaptation," in *Eurospeech*, Geneva, 2003, pp. 1981–1984.

[23] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1997.

[24] D. Povey and G. Saon, "Feature and model space speaker adaptation with full covariance gaussians," in *Interspeech*, Pittsburgh, 2006, pp. 1145–1148.

[25] J. Ganitkevitch, "Speaker adaptation using maximum likelihood linear regression," Rheinisch-Westfälische Technische Hochschule Aachen, Tech. Rep., 2005.

[26] L. Uebel and P. Woodland, "Improvements in linear transform based speaker adaptation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Salt Lake City, 2001, pp. 49–52.

[27] L. Wang and P. Woodland, "MPE-based discriminative linear transform for speaker adaptation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Montreal, 2004, pp. 321–324.

[28] C. J. Leggeter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaption of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[29] A. Stolcke, S. S. Kajarekar, L. Ferrer, and E. Shriberg, "Speaker recognition with session variability normalization based on MLLR adaptation transforms," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1987–1998, 2007.

[30] M. Gales, "The generation and use of regression class trees for MLLR adaptation," Cambridge University Engineering Department, Tech. Rep., 1996.

[31] S. Cheng, Y.-Y. Xu, H.-M. Wang, and H.-C. Fu, "Automatic construction of regression class tree for MLLR via model-based hierarchical clustering," *Lecture Notes in Computer Science*, vol. 4274, pp. 390–398, 2006.

[32] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *Computer Jurnal*, vol. 41, pp. 578–588, 1998.

[33] S. M. Ahadi and P. C. Woodland, "Combined Bayesian and predictive techniques next term for previous termrapid speaker adaptationnext term of continuous density hidden Markov models," *Computer Speech and Language*, vol. 11, no. 3, pp. 187–206, 1997.

[34] L. He, J. Wu, D. Fang, and W. Wu, "Speaker adaptation based on combination of MAP estimation and weighted neighbor regression," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Istanbul, 2000, pp. 98–984.

[35] P. Červa, "Řízená a neřízená adaptace na mluvčího v systémech rozpoznávání řeči," Ph.D. dissertation, Technická univerzita v Liberci, Fakulta mechatroniky a mezioborových inženýrských studií, 2007.

[36] K. Shinoda and C.-H. Lee, "Structural MAP speaker adaptation using hierarchical priors," in *IEEE Automatic Speech Recognition and Understanding*, Santa Barbara, 1997, pp. 381–388.

[37] T. Andre, M. Olivier, S. Chin-Hui, and L. W. Chou, "Structural maximum a posteriori linear regression for unsupervised speaker adaptation," in *IEEE International Conference on Speech and Language Processing*, Beijing, 2000, pp. 256–259.

[38] G. Jang, S. Woo, M. Jin, and C. D. Yoo, "Improvements in speaker adaptation using weighted training," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Hong Kong, 2003, pp. 548–551.

[39] J. Takahashi and S. Sagayama, "Vector-field-smooted Bayesian learning for incremental speaker adaptation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Atlanta, 1996, pp. 696–699.

[40] M. Tonomura, T. Kosaka, and S. Mutsunaga, "Speaker adaptation based on transfer vector field smoothing using maximum a posteriori probability estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Detroit, 1995, pp. 688–691.

[41] W. Chou, "Maximum a posterior linear regression with elliptically symmetric matrix variate priors," in *Eurospeech*, vol. 1, Budapest, 1999, pp. 1–4.

[42] X. Lei, J. Hamaker, and X. He, "Robust feature space adaptation for telephony speech recognition," in *IEEE International Conference on Spoken Language Processing*, Pittsburgh, 2006, pp. 773–776.

[43] M. Padmanabhan, L. Bahl, D. Nahamoo, and M. Picheny, "Speaker clustering and transformation for speaker adaptation in speech recognition systems," *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 71–77, 1998.

[44] A. Sankar, F. Beaufays, and V. Digalakis, "Training data clustering for improved speech recognition," in *Eurospeech*, Madrid, 1995, pp. 502–505.

[45] C. Huang, T. Chen, and E. Chang, "Adaptive model combination for dynamic speaker selection training," in *IEEE International Conference on Spoken Language Processing*, vol. 1, Denver, 2002, pp. 774–777.

[46] M. Morishima, T. Isobe, and J. Takahashi, "Phonetically adaptive cepstrum mean normalization for acousticmismatch compensation," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, 1997, pp. 436–441.

[47] G. Saon, A. Dharanipragada, and D. Povey., "Feature space gaussianization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, Montreal, 2004, pp. 329–332.

[48] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *IEEE International Conference on Spoken Language Processing*, Philadelphia, 1996, pp. 1137–1140.

[49] S. Matsoukas, R. Schwartz, H. Jin, and L. Nguyen, "Practical implementations of speaker-adaptive training," in *DARPA Speech Recognition Workshop*, Virginia, 1997.

[50] M. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 417–428, 2000.

[51] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Munich, 1997, pp. 1039–1042.

[52] L. Lee and R. Rose, "Speaker normalization using efficient frequency warping procedures," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Atlanta, 1996, pp. 353–356.

[53] S. Tsakalidis, V. Doumpiotis, and W. Byrne, "Discriminative linear transforms for feature normalization and speaker adaptation in HMM estimation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 367–376, 2005.

[54] L. Wang and P. Woodland, "Discriminative adaptive training using the mpe criterion," in *IEEE Automatic Speech Recognition and Understanding*, Virgin Islands, 2003, pp. 279–284.

[55] M. J. F. Gales, "Multiple-cluster adaptive training schemes," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Salt Lake City, 2001, pp. 361–364.

[56] K. Yu and M. J. F. Gales, "Discriminative cluster adaptive training," in *IEEE International Conference on Spoken Language Processing*, vol. 14, no. 5, Pittsburgh, 2006, pp. 1694–1703.

[57] D. Paczolay, A. Kocsor, and L. Tóth, "Real-time vocal tract length normalization in a phonological awareness teaching system," *Lecture Notes in Computer Science*, vol. 2807, pp. 309–314, 2003.

[58] J. W. McDonough, "Speaker compensation with all-pass transforms," Ph.D. dissertation, Johns Hopkins University, Baltimore, Maryland, 2000.

[59] S. Panchapagesan and A. Alwan, "Multi-parameter frequency warping for VTLN by gradient search," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, Toulouse, 2006, pp. 1181–1184.

[60] M. Westphal, T. Schultz, and A. Waibel, "Linear discriminant a new criterion for speaker normalization," in *IEEE International Conference on Spoken Language Processing*, Sydney, 1998, pp. 827–830.

[61] P. Červa, K. Paleček, J. Silovský, and J. Nouza, "An investigation into VTLN for improved transcription of Czech broadcast programs," in *IEEE International Symposium ELMAR*, Zadar, 2011, pp. 201–204.

[62] J. Lööf, H. Ney, and S. Umesh, "VTLN warping factor estimation using accumulation of sufficient statistics," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, 2006, pp. 1201 – 1204.

[63] S. Umesh, A. Zolnay, and H. Ney, "Implementing frequency-warping and VTLN through linear transformation of conventional MFCC," in *Interspeech*, Lisboa, 2005, pp. 269–272.

[64] M. Pitz, "Investigations on linear transformations for speaker adaptation and normalization," Ph.D. dissertation, Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule, Aachen, 2005.

[65] D. R. Sanand, D. D. Kumar, and S. Umesh, "Linear transformation approach to VTLN using dynamic frequency warping," in *Interspeech*, Antwerp, 2007, pp. 1138–1141.

[66] S. Panchapagesan and A. Alwan, "Frequency warping for VTLN and speaker adaptation by linear transformation of standard MFCC," *Computer Speech and Language*, vol. 23, pp. 42 – 64, 2008.

[67] X. Cui and A. Alwan, "Adaptation of children's speech with limited data based on formant-like peak alignment," *Computer Speech and Language*, vol. 20, pp. 400–419, 2006.

[68] L. Machlica, Z. Zajíc, and A. Pražák, "Methods of unsupervised adaptation in online speech recognition," in *Specom*, St. Petersburg, 2009, pp. 448–453.

[69] L. Uebel and P. Woodland, "Speaker adaptation using lattice-based MLLR," in *ISCA Tutorial and Research Workshop on Adaptation Methods for Speech Recognition*, Sophia Antipolis, 2001, pp. 57–60.

[70] M. Padmanabhan, G. Saon, and G. Zweig, "Lattice-based unsupervised MLLR for speaker adaptation," in *ISCA ITRW ASR*, Paris, 2000, pp. 128–131.

[71] Y. Li, H. Erdogan, Y. Gao, and E. Marcheret, "Incremental on-line feature space MLLR adaptation for telephony speech recognition," in *IEEE International Conference on Spoken Language Processing*, Denver, 2002, pp. 1417–1420.

[72] P. Fischerová, "Detekce změny řečníka v řečovém signálu," Ph.D. dissertation, Západočeská univerzita v Plzni, Fakulta aplikovaných Věd, Katedra kybernetiky, 2007.

[73] J. Žďánský, "Metody detekce změny mluvčího v akustickém signálu," Ph.D. dissertation, Technická univerzita v Liberci, Fakulta mechatroniky a mezioborových inženýrských studií, 2005.

[74] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, pp. 1437–1462, 1997.

[75] J. Tatarinov and P. Pollák, "HMM and EHMM based voice activity detectors and design of testing platform for VAD classification," in *Digital Technologies*, vol. 1, Žilina, 2008, pp. 1–4.

[76] Z.-P. Zhang and S. F. K. Ohtsuki, "On-line incremental speaker adaptation with automatic speaker change detection," in *Proceedings of the Acoustics, Speech, and Signal Processing*, vol. 2, Istanbul, 2000, pp. 961–964.

[77] Z. Zhang and S. Furui, "An online incremental speaker adaptation method using speaker-clustered initial models," in *IEEE International Conference on Spoken Language Processing*, vol. 3, Beijing, 2000, pp. 694–697.

[78] J. H. H. Rongqing Huang, "Advances in unsupervised audio segmentation for the broadcast news and NGSW corpora," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Montreal, 2004, pp. 741–744.

[79] H. Gish, M.-H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Toronto, 1991, pp. 873 –876.

[80] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *DARPA Broadcast News Transcription and Understanding Workshop*, Virginia, 1998, pp. 127–132.

[81] J. Ajmera, I. McCowan, and H. Bourlard, "Robust speaker change detection," *IEEE Signal Processing Letters*, vol. 11, pp. 649–651, August 2004.

[82] M. Kotti, E. Benetos, L. Gustavo, and P. M. Martins, "Speaker change detection using BIC: A comparison on two datasets." in *International Symposium on Communications, Control and Signal Processing*, Marrakech, 2006.

[83] B. Zhou and J. H. Hansen, "Efficient audio stream segmentation via the combined T2 statistic and Bayesian information criterion," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 4, pp. 467–474, 2005.

[84] D. Giuliani and F. Brugnara, "Acoustic model adaptation with multiple supervisions," in *TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, 2006, pp. 151—154.

[85] J. Lööf, C. Gollan, and H. Ney, "Speaker adaptive training using shift-MLLR," in *Interspeech*, Brisbane, 2008, pp. 1701–1705.

[86] A. Gunawardana and W. Byrne, "Discounted likelihood linear regression for rapid speaker adaptation," *Computer Speech and Language*, vol. 15, pp. 15–38, 2001.

[87] P. Červa, J. Nouza, and J. Silovský, "Two-step unsupervised speaker adaptation based on speaker and gender recognition and HMM combination," in *Interspeech*, Pittsburgh, 2006, pp. 2326–2329.

[88] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, 2004.

[89] Z. Zajíc, "Metody normalizace skóre v úloze verifikace řečníka," Master's thesis, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky, 2006.

[90] S. Yoshizawa, A. Baba, K. Matsunami, Y. Mera, M. Yamada, and K. Shikano, "Unsupervised speaker adaptation based on sufficient HMM statistics of selected speakers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, 2001, pp. 341–344.

[91] R. Gomez, T. Toda, H. Saruwatari, and K. Shikano, "Improving rapid unsupervised speaker adaptation based on HMM sufficient statistics," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, 2006, pp. 1001–1004.

[92] C. Huang, T. Chen, and E. Chang, "Transformation and combination of hidden Markov models for speaker selection training." International Speech Communication Association, 2004.

[93] C. Breslin, K. Chin, M. Gales, K. Knill, and H. Xu, "Prior information for rapid speaker adaptation," in *Interspeech*, Chiba, 2010, pp. 1644–1647.

[94] M. Gales and R. van Dalen, "Predictive linear transforms for noise robust speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU2007)*, Kyoto, 2007, pp. 59–64.

[95] R. Westwood, "Speaker adaptation using eigenvoices," Cambridge University Engineering Department, Tech. Rep., 1999.

[96] R. Kuhn, P. Nguyen, J.-C. Junqua, and L. Goldwasser, "Eigenfaces and eigenvoices: Dimensionality reduction for specialized pattern recognition," in *IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, 1998, pp. 71–76.

[97] I. Jolliffe, *Principal Component Analysis*. Springer Series in Statistics, 2002.

[98] G. H. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis 2*, vol. 2, pp. 205–224, 1965.

[99] K. Chen, W. Liau, H. Wang, and L. Lee, "Fast speaker adaptation using eigenspace-based maximum likelihood linear regression," in *IEEE International Conference on Spoken Language Processing*, vol. 3, Beijing, 2000, pp. 742–745.

[100] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation," in *IEEE International Conference on Spoken Language Processing*, Sydney, 1998, pp. 1771–1774.

[101] B. Mak, J. T. Kwok, and S. Ho, "Kernel eigenvoice speaker adaptation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 984–992, 2005.

[102] P. Kenny, M. Mihoubi, and P. Dumouchel, "New MAP estimators for speaker recognition," in *EUROSPEECH*, Geneva, 2003, pp. 2961–2964.

[103] P. Kenny, G. Boulianne, and P. Dumouchel, "Maximum likelihood estimation of eigenvoices and residual variances for large vocabulary speech recognition tasks," in *IEEE International Conference on Spoken Language Processing*, Denver, 2002, pp. 57–60.

[104] B. Mak and R. Hsiao, "Improving eigenspace-based MLLR adaptation by kernel PCA," in *IEEE International Conference on Spoken Language Processing*, vol. 1, Jeju Island, 2004, pp. 13–16.

[105] K. Chen and H. Wang, "Eigenspace-based maximum a posteriori linear regression for rapid speaker adaptation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Salt Lake City, 2001, pp. 317–320.

[106] R. L. Gorsuch, *Factor Analysis. Second edition*, N. L. E. A. Hillsdale, Ed. Psychology Press, 1983.

[107] R. L. Gorsuch, "Common factor analysis versus component analysis: Some well and little known facts," *Multivariate Behavioral Research*, vol. 25, pp. 33–39, 1990.

[108] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transaction on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[109] L. Machlica and Z. Zajíc, "Analysis of the influence of speech corpora in the PLDA verification in the task of speaker recognition," *Lecture Notes in Computer Science*, vol. 7499, pp. 464–471, 2012.

[110] L. Burget, N. Brümmer, D. Reynolds, P. Kenny, J. Pelecanos, R. Vogt, F. Castaldo, N. Dehak, R. Dehak, O. Glembek, Z. Karam, J. J. Noecker, Y. H. Na, C. C. Costin, V. Hubeika, S. Kajarekar, N. Scheffer, and J. Černocký, "Robust speaker recognition over varying channels," Johns Hopkins University CLSP Summer Workshop, Tech. Rep., 2008.

[111] K. Visweswariah, V. Goel, and R. Gopinath, "Structuring linear transforms for adaptation using training time information," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Orlando, 2002, pp. 585–588.

[112] D. Povey and K. Yao, "A basis representation of constrained MLLR transforms for robust adaptation," *Computer Speech and Language*, vol. 26, no. 1, pp. 35–51, 2012.

[113] K. Visweswariah, V. Goel, and R. Gopinath, "Maximum likelihood training of bases for rapid adaptation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, 2002, pp. 585–588.

[114] J. A. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms.* Springer Publishing, 2005.

[115] J. Trmal, "Spatio-temporal structure of feature vectors in neural network adaptation," Ph.D. dissertation, Faculty of Applied Sciences, University of West Bohemia, 2011.

[116] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 5, no. 4, pp. 115–133, 1943.

[117] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1996.

[118] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, pp. 1190–1208, 1994.

[119] C. Igel and M. Hüsken, "Improving the rprop learning algorithm," in *International Symposium on Neural Computation*, Berlin, 2000, pp. 115–121.

[120] E. Parviainen, "Dimension reduction for regression with bottleneck neural networks," *Lecture Notes in Computer Science*, vol. 6283, pp. 37–44, 2010.

[121] V. Radová and P. Vopálka, "Methods of sentences selection for read-speech corpus design," *Lecture Notes in Computer Science*, vol. 1692, pp. 165–170, 1999.

[122] P. Pollák, J. Černocký, J. Boudy, K. Choukri, H. van den Heuvel, K. Vicsi, A. Virag, R. Siemund, W. Majewski, J. Sadowski, P. Staroniewicz, H. Tropf, J. Kochanina, A. Ostroukhov, M. Rusko, and M. Trnka, "SpeechDat(E) - eastern european telephone speech databases," in *XLDB - Very Large Telephone Speech Databases*. Paris: European Language Recources Association, 2000.

[123] J. Psutka, *Komunikace s počítačem česky*. ACADEMIA Praha, 1995.

[124] J. Kanis, "Statistický automatický překlad čeština - znakovaná řeč," Ph.D. dissertation, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky, 2009.

[125] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, vol. 14, Montreal, 1995, pp. 1137–1143.

[126] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and their Application (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, 1997.

[127] S. Riezler and J. T. Maxwell, "On some pitfalls in automatic evaluation and significance testing for MT," in *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, 2005, pp. 57—-64.

[128] L. Machlica and Z. Zajíc, "The speaker adaptation of an acoustic model," in *The 1st Young Researchers Conference on Applied Sciences*, Pilsen, 2007, pp. 212–217.

[129] Z. Zajíc, L. Machlica, and L. Müller, "Refinement approach for adaptation based on combination of MAP and fMLLR," *Lecture Notes in Computer Science*, vol. 5729, pp. 274–281, 2009.

[130] L. Machlica, Z. Zajíc, and L. Müller, "Discriminative adaptation based on fast combination of DMAP and DfMLLR," in *Interspeech*, Chiba, 2010, pp. 534–537.

[131] A. Pražák, L. Müller, J. V. Psutka, and J. Psutka, "Live TV subtitling - fast 2-pass LVCSR system for online subtitling," in *Sigmap*, Lisabon, 2007, pp. 139–142.

[132] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *ICSLP*, Denver, 2002, pp. 901–904.

[133] A. Pražák, Z. Zajíc, L. Machlica, and J. V. Psutka, "Fast speaker adaptation in automatic online subtitling," in *SIGMAP*, Milan, 2009, pp. 126–130.

[134] Z. Zajíc, L. Machlica, and L. Müller, "Robust statistic estimates for adaptation in the task of speech recognition," *Lecture Notes in Computer Science*, vol. 6231, pp. 464–471, 2010.

[135] R. Gomez, T. Toda, H. Saruwatari, and K. Shikano, "Rapid unsupervised speaker adaptation using single utterance based on MLLR and speaker selection," in *Interspeech*, Antwerp, 2007, pp. 262–265.

[136] Z. Zajíc, L. Machlica, and L. Müller, "Initialization of fMLLR with sufficient statistics from similar speakers," *Lecture Notes in Computer Science*, vol. 6836, pp. 187–194, 2011.

[137] Z. Zajíc, L. Machlica, and L. Müller, "Initialization of adaptation by sufficient statistics using phonetic tree," in *IEEE International Conference on Signal Processing*, Beijing, 2012, (in press).

[138] Z. Zajíc, L. Machlica, and L. Müller, "Robust adaptation techniques dealing with small amount of data," *Lecture Notes in Computer Science*, vol. 7499, pp. 418–487, 2012.

[139] P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Elsevier, 2010, iSBN 978-0-12-374726-6.

[140] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analysers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[141] Z. Zajíc, L. Machlica, and L. Müller, "Bottleneck ANN: Dealing with small amount of data in shift-MLLR adaptation," in *IEEE International Conference on Signal Processing*, Beijing, 2012, (in press).

Příloha A

Nastavení adaptačních metod

τ	5	10	16	20
Acc	71,54	72,99	73,09	72,95

Tabulka A.1: Výsledky (Acc[%]) metody MAP v na nastavení jejího parametru τ, CzT korpus.

vnitřní iterace	2	5	10	15	20	25	30
Acc	76,31	76,40	76,59	76,60	76,94	76,97	76,75

Tabulka A.2: Výsledky (Acc[%]) metody fMLLR v závyslosti na počtu vnitřních iterací k odhadu transformační matice, CzT korpus.

iterace	1	2	3	4
Acc	76,94	76,95	77,02	76,91

Tabulka A.3: Výsledky (Acc[%]) metody fMLLR s okupačním prahem pro regresní strom $Th = 1000$ pro více iterací celé metody, CzT korpus.

Příloha B

Tabulky výsledků

počet vět	SI	MAP-16	MLLR-global	MLLR-1000	fMLLR-global	fMLLR-1000	VTNL-100
1	65,32	66,83	70,16	70,16	70,42	70,42	66,43
2	65,32	67,93	72,01	71,93	74,58	74,48	66,79
3	65,32	68,14	71,99	72,56	74,65	74,48	66,50
4	65,32	68,16	71,93	72,26	74,84	75,00	66,39
5	65,32	68,35	71,71	72,75	74,90	74,65	66,60
6	65,32	69,18	72,23	72,78	74,90	75,13	66,81
8	65,32	70,03	72,49	73,02	74,90	75,29	66,84
10	65,32	70,61	72,30	73,81	75,00	75,54	66,86
12	65,32	70,66	72,45	74,31	74,87	75,78	66,67

Tabulka B.1: Výsledky (Acc[%]) adaptačních metod při různém počtu adaptačních vět, pro korpus CzT. V metodě MAP-16 byly adaptovány střední hodnoty, kovarianční matice i váhy složek najednou. Konstanta τ byla experimentálně nastavena na hodnotu 16. Regresní strom v metodě (f)MLLR-1000 byl konstruován s 32 listovými uzly s okupačním prahem $Th = 1000$. Metoda (f)MLLR-global odhadovala pouze jednu globální transformaci. Metoda VTLN využívala regresní strom s 64 listovými uzly s okupačním prahem $Th = 100$. SI označuje neadaptovaný model.

počet vět	SI	fMLLR-global	fMLLR-1000
1	68,75	14,04	14,04
2	68,75	56,36	56,12
3	68,75	66,74	64,63
4	68,75	69,58	64,66
5	68,75	70,23	66,53
6	68,75	70,74	67,11
8	68,75	72,30	71,03
10	68,75	72,30	71,46
12	68,75	72,33	73,02

Tabulka B.2: Výsledky (Acc[%]) adaptačních metod při různém počtu adaptačních vět, pro korpus SD-E. Regresní strom v metodě fMLLR-1000 byl konstruován s 32 listovými uzly s okupačním prahem $Th = 1000$. Metoda fMLLR-global odhadovala pouze jednu globální transformaci. SI označuje neadaptovaný model.

počet vět	SI	fMLLR-1000	fMLLR-global	fMLLR-1000-Th_γ0,5	fMLLR-1000-Th_γ0,3
1	68,75	14,04	14,04	69,15	69,92
2	68,75	56,12	56,36	70,69	70,92
3	68,75	64,63	66,74	70,98	71,13
4	68,75	64,66	69,58	70,80	70,68
5	68,75	66,53	70,23	70,53	70,17
6	68,75	67,11	70,74	70,35	70,04
8	68,75	71,03	72,30	71,24	71,33
10	68,75	71,46	72,30	71,09	71,52
12	68,75	73,02	72,33	72,57	72,65

Tabulka B.3: Výsledky (Acc[%]) metody fMLLR při vyuřití metody zrobustnění statistik, pro korpus SD-E. Metoda fMLLR-1000 využívá regresní strom s okupačním prahem $Th = 1000$. Metoda fMLLR-global odhaduje pouze jednu globální transformaci. fMLLR-Th_γ je označení fMLLR adaptace s ruznou volbou prahu Th_γ pro relevanci adaptacních statistik. SI označuje neadaptovaný model.

počet vět	SI model	fMLLR-global	fMLLR-1000	fMLLR-1000 -Inic	fMLLR-1000 -Nbest	fMLLR-1000 -Nbest-phone
1	68,75	14,04	14,04	69,29	68,30	69,85
2	68,75	56,36	56,12	69,89	68,16	69,99
3	68,75	66,74	64,63	69,78	68,96	70,00
4	68,75	69,58	64,66	70,09	69,05	70,21
5	68,75	70,23	66,53	69,87	69,30	70,29
6	68,75	70,74	67,11	69,31	69,53	71,02
8	68,75	72,30	71,03	71,41	70,52	71,44
10	68,75	72,30	71,46	72,54	70,50	71,65
12	68,75	72,33	73,02	72,83	70,89	71,76

Tabulka B.4: Výsledky (Acc[%]) metody fMLLR pro různé principy inicializace statistik, pro korpus SD-E. Metoda fMLLR-1000 využívá regresní strom s okupačním prahem $Th = 1000$. Metoda fMLLR-global odhaduje pouze jednu globální transformaci. fMLLR-Inic je označení fMLLR adaptace s inicializací statistikami z SI modelu, fMLLR-Nbest s inicializací N nejbližšími řečníky z trénovací databáze a fMLLR-Nbest-phone s inicializací N nejbližšími řečníky s využitím fonetické informace. SI označuje neadaptovaný model.

počet vět	SI	fMLLR-global	ML	EV	FA	ICA	Wnode
1	68,75	14,04	70,59	69,28	69,20	69,63	69,19
2	68,75	56,36	71,01	69,78	69,45	69,93	68,39
3	68,75	66,74	71,46	69,88	69,37	69,83	68,30
4	68,75	69,58	71,21	69,83	69,46	69,85	69,29
5	68,75	70,23	71,42	69,81	69,41	69,90	69,30
6	68,75	70,74	71,63	69,89	69,51	69,97	69,22
8	68,75	72,30	71,87	70,29	69,50	70,13	69,27
10	68,75	72,30	72,15	70,47	69,47	70,20	69,42
12	68,75	72,33	72,14	70,46	69,46	70,11	69,33

Tabulka B.5: Výsledky (Acc[%]) pro lineární kombinaci různých bázových matic, pro korpus SD-E. Wnode označuje bázi danou maticemi shluků trénovacích řecníků, FA je daná faktorovou analýzou, ICA určená z analýzy nezávislých komponent, ML odhad vycházející z ML kritéria a EV definováno největšími vlastními vektory. Pro porovnání jsou uvedeny výsledky fMLLR globální adaptace a výsledky s neadaptovaným SI modelem.

počet vět	SI	fMLLR-1000	shiftMLLR-100	ann-shiftMLLR-100
1	68,75	14,04	69,84	71,00
2	68,75	56,12	70,56	71,21
3	68,75	64,63	70,79	71,35
4	68,75	64,66	70,94	70,74
5	68,75	66,53	71,16	71,06
6	68,75	67,11	71,31	71,16
8	68,75	71,03	71,76	71,27
10	68,75	71,46	72,14	71,26
12	68,75	73,02	71,54	71,05

Tabulka B.6: Výsledky (Acc[%]) adaptace shiftMLLR s využitím ANN pro redukci dimenze vektoru, pro SD-E korpus. Pro porovnání uvedeny i výsledky adaptace shiftMLLR (globální i s regresním stromem s okupačním prahem $Th = 1000$. SI označuje neadaptovaný model.

Seznam publikovaných prací

1. L. Machlica and Z. Zajíc, "The speaker adaptation of an acoustic model," in *The 1st Young Researchers Conference on Applied Sciences*, Pilsen, 2007, pp. 212–217.
2. Z. Zajíc, J. Vaněk, L. Machlica, and A. Padrta, "A cohort methods for score normalization in speaker verification system, acceleration of on-line cohort methods," in *Specom*, Moskow, 2007, pp. 367–372.
3. Z. Zajíc, L. Machlica, A. Padrta, J. Vaněk, and V. Radová, "An expert system in speaker verification task," in *Interspeech*, vol. 9, Brisbane, 2008, pp. 355–358.
4. Z. Zajíc, L. Machlica, and L. Müller, "Refinement approach for adaptation based on combination of MAP and fMLLR," *Lecture Notes in Computer Science*, vol. 5729, pp. 274–281, 2009.
5. L. Machlica, Z. Zajíc, and A. Pražák, "Methods of unsupervised adaptation in online speech recognition," in *Specom*, St. Petersburg, 2009, pp. 448–453.
6. A. Pražák, Z. Zajíc, L. Machlica, and J. V. Psutka, "Fast speaker adaptation in automatic online subtitling," in *SIGMAP*, Milan, 2009, pp. 126–130.
7. L. Machlica, Z. Zajíc, and L. Müller, "Discriminative adaptation based on fast combination of DMAP and DfMLLR," in *Interspeech*, Chiba, 2010, pp. 534–537.
8. Z. Zajíc, L. Machlica, and L. Müller, "Robust statistic estimates for adaptation in the task of speech recognition," *Lecture Notes in Computer Science*, vol. 6231, pp. 464–471, 2010.
9. L. Machlica, J. Vaněk, and Z. Zajíc, "Fast estimation of gaussian mixture model parameters on GPU using CUDA," in *International Conference on Parallel and Distributed Computing, Applications and Technologies*, Gwangju, 2011, pp. 167–172.
10. Z. Zajíc, L. Machlica, and L. Müller, "Initialization of fMLLR with sufficient statistics from similar speakers," *Lecture Notes in Computer Science*, vol. 6836, pp. 187–194, 2011.
11. L. Machlica and Z. Zajíc, "Factor analysis and nuisance attribute projection revisited," in *Interspeech*, Portland, 2012, (in press).
12. L. Machlica and Z. Zajíc, "Analysis of the influence of speech corpora in the PLDA verification in the task of speaker recognition," *Lecture Notes in Computer Science*, vol. 7499, pp. 464–471, 2012
13. Z. Zajíc, L. Machlica, and L. Müller, "Robust adaptation techniques dealing with small amount of data," *Lecture Notes in Computer Science*, vol. 7499, pp. 418–487, 2012.
14. Z. Zajíc, L. Machlica, and L. Müller, "Bottleneck ANN: Dealing with small amount of data in shift-MLLR adaptation," in *IEEE International Conference on Signal Processing*, Beijing, 2012, (in press).
15. Z. Zajíc, L. Machlica, and L. Müller, "Initialization of adaptation by sufficient statistics using phonetic tree," in *IEEE International Conference on Signal Processing*, Beijing, 2012, (in press).