# Západočeská univerzita v Plzni

## Fakulta aplikovaných věd

Katedra kybernetiky


Diplomová práce

# Optimalizovaná a paralelní implementace metod pro rozpoznávání řeči


Marie Kunešová



Plzeň 2013

# Prohlášení

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

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

V Plzni dne 19. května 2013

.....

vlastnoruční podpis

# Poděkování

Na tomto místě bych ráda poděkovala svému vedoucímu práce, panu Ing. Janu Vaňkovi, PhD., za věnovaný čas při konzultacích a poskytnuté rady a připomínky při zpracování diplomové práce.

# Anotace

Tato práce se zabývá analýzou a optimalizací metod rozpoznávání řeči z hlediska rychlosti výpočtu a přesnosti rozpoznávání. Je použito rozpoznávání řeči využívající skrytých Markovových modelů. Optimalizace z hlediska času výpočtu je zaměřena na zrychlení výpočtu akustických pravděpodobností a dekódování jako celku. Druhá část práce se zabývá analýzou a optimalizací rozpoznávání z hlediska jeho úspěšnosti. Rozpoznávány byly promluvy sestávající z posloupnosti číslovek. Je stanoven optimální počet stavů HMM připadajících na každé slovo a způsob normalizace dat a je zavedena penalizace slov. Dále je získána referenční posloupnost stavů HMM a na jejím základě jsou hledány nejčastěji se vyskytující záměny. Závěrem práce je zkoumán vliv zkreslení dat na kvalitu rozpoznávání.

*Klíčová slova*

Rozpoznávání řeči, HMM, algoritmus forward-backward, Viterbiův algoritmus, MFCC

# Annotation

This thesis concerns the analysis and optimization methods of speech recognition in regards to computation speed and accuracy of recognition. Speech recognition is done with the use of Hidden Markov Models. Optimization in regards to computation time is targeted at improving the speed of acoustic likelihood computation and decoding as a whole. The second part of the thesis is concerning the analysis and optimization of speech recognition in regards to accuracy. Sequences of digits from 0 to 9 were being recognized. The optimal number of HMM states for each word and the approach to data normalization were determined and word penalization was applied. In the conclusion of the thesis, the influence of noise on the quality of recognition is examined.

*Key Words*

Speech recognition, HMM, forward-backward algorithm, Viterbi algorithm, MFCC

# Obsah

**1 Úvod** **11**
1.1 Cíle diplomové práce . . . . . 11

**2 Rozpoznávání řeči** **12**
2.1 Statistický přístup k rozpoznávání řeči . . . . . 12

**3 Parametrizace řečového signálu** **14**
3.1 Význam parametrizace . . . . . 14
3.2 Metody krátkodobé analýzy . . . . . 14
3.3 Metoda melovských kepstrálních koeficientů . . . . . 15
3.3.1 Preemfáze . . . . . 15
3.3.2 Hammingovo okénko . . . . . 15
3.3.3 Popis metody MFCC . . . . . 16
3.3.4 Delta a delta-delta koeficienty . . . . . 17
3.4 Perceptivní lineární prediktivní analýza . . . . . 18

**4 Akustické a jazykové modelování** **20**
4.1 Akustické modelování . . . . . 20
4.1.1 Struktura skrytých Markovových modelů . . . . . 20
4.1.2 Pravděpodobnost generování promluvy modelem . . . . . 21
4.1.3 Algoritmus forward-backward . . . . . 22
4.1.4 Viterbiův algoritmus . . . . . 24
4.1.5 Trénování parametrů skrytého Markovova modelu . . . . . 25
4.2 Jazykové modelování . . . . . 27

**5 Dekódování** **28**
5.1 Definice úlohy dekódování . . . . . 28

5.2 Časově synchronní Viterbiovo prohledávání . . . . . 29

**6 Optimalizace úlohy rozpoznávání řeči z hlediska času výpočtu 31**

6.1 Zrychlení výpočtu akustických pravděpodobností . . . . . 31

6.1.1 Převod výpočtu na kombinaci dvou jednodušších operací . . . . . 31

6.1.2 Aproximace výpočtu funkce *addLog* . . . . . 33

6.1.3 Volba parametrů . . . . . 34

6.1.4 Výsledky . . . . . 35

6.2 Zrychlení dekódování pomocí prořezávání . . . . . 36

**7 Optimalizace z hlediska úspěšnosti rozpoznávání 38**

7.1 Použitá data a modely . . . . . 38

7.1.1 Popis nahrávek . . . . . 38

7.1.2 Popis použitých modelů . . . . . 38

7.2 Hodnocení úspěšnosti rozpoznávání . . . . . 39

7.3 Určení počtu stavů připadajících na každé slovo . . . . . 41

7.4 Normalizace příznakových vektorů . . . . . 43

7.5 Penalizace slov . . . . . 44

7.5.1 Stanovení penalty na základě úspěšnosti rozpoznávání . . . . . 44

7.5.2 Stanovení penalty na základě statistik z HResults . . . . . 45

7.5.3 Stanovení penalty na základě počtu rozpoznaných slov . . . . . 47

**8 Analýza výsledků rozpoznávání 48**

8.1 Vyhodnocení z hlediska konkrétních stavů HMM . . . . . 48

8.1.1 Srovnání s referenční posloupností stavů . . . . . 48

8.1.2 Určení nejčastěji zaměňovaných stavů . . . . . 50

8.2 Srovnání středních hodnot a variancí modelu a dat . . . . . 52

8.3 Vliv zkreslení dat na kvalitu rozpoznávání . . . . . 54

8.3.1 Přidání šumu s různou variancí k příznakovým vektorům . . . . . 54

8.3.2 Přidání šumu s různou střední hodnotou k příznakovým vektorům . 55

8.3.3 Zkreslení samotných nahrávek . . . . . 58

**9 Závěr 60**

# Seznam obrázků


2.1 Blokové schéma systému rozpoznávání řeči . . . . . 13

3.1 Banka trojúhelníkových filtrů v melovské a původní škále . . . . . 16

4.1 Skrytý Markovův model s pěti stavy . . . . . 21

6.1 Průměrná ponechaná část z matice hodnot $\alpha$ v algoritmu forward-backward 37

7.1 Schématické znázornění jazykového modelu . . . . . 39
7.2 Celková úspěšnost rozpoznávání slov pro různé velikosti penalty vložení a různé počty stavů HMM připadající na jedno slovo . . . . . 42
7.3 Celková úspěšnost rozpoznávání slov pro různé velikosti penalty vložení a různé počty stavů HMM připadající na jedno slovo - detail . . . . . 42
7.4 Úspěšnost rozpoznávání pro různá nastavení normalizace dat . . . . . 43
7.5 Závislost celkové úspěšnosti rozpoznávání slov na zvolené výši penalizace přechodu modelu do stavů nového slova . . . . . 45
7.6 Počty zaměněných, chybějících a přebývajících slov v rozpoznané posloupnosti slov oproti referenčnímu přepisu, v závislosti na zvolené penaltě . . . 46
7.7 Počty rozpoznaných slov pro různé hodnoty penalty a jejich skutečný počet 47

8.1 Srovnání nejpravděpodobnější posloupnosti stavů pro jednu z vět, určené dekódováním s referenčním přepisem a bez něj . . . . . 49
8.2 Relativní podíly časových okamžiků, odpovídajících konkrétnímu stavu modelu, které byly rozpoznány jako stav jiný . . . . . 50
8.3 Srovnání výstupních hustot pravděpodobností dvou stavů HMM a dat . . . 51
8.4 Závislost úspěšnosti rozpoznávání na varianci přidaného šumu pro zašuměná trénovací, testovací a trénovací i testovací data, průměry ze tří realizací . . 55

8.5 Závislost úspěšnosti rozpoznávání na střední hodnotě přidaného šumu pro zašuměná trénovací, testovací a trénovací i testovací data . . . . . 57

# Seznam tabulek

6.1 Srovnání časů a přesností výpočtu 50 milionů hodnot pro různé varianty . 35

7.1 Úspěšnost rozpoznávání pro různé počty stavů připadajících na jedno slovo 41

8.1 Přehled indexů stavů náležejících jednotlivým slovům . . . . . 52
8.2 Srovnání variancí modelu a dat pro přidaný šum s nulovou střední hodnotou a různou variancí . . . . . 56
8.3 Srovnání středních hodnot modelu a dat pro data zkreslená posunem o konstantní hodnotou . . . . . 57
8.4 Úspěšnost rozpoznávání pro různé kombinace zkreslených dat . . . . . 59
8.5 Srovnání variancí testovacích dat a modelu . . . . . 59
8.6 Srovnání středních hodnot testovacích dat a modelu . . . . . 59

# 1 Úvod

Problematika rozpoznávání řeči se zabývá převodem mluvené řeči na text. Rozpoznávání řeči je využíváno v řadě praktických aplikací, které zahrnují například hlasové ovládání strojů, diktování textu, telefonní aplikace nebo zpracování informací obsažených v řečových nahrávkách či automatické titulkování.

Úspěšné rozpoznávání řeči je ovlivněno množstvím faktorů. Různí lidé mají odlišné hlasy a mohou stejná slova vyslovovat různým způsobem. Lišit se může mimo jiné barva hlasu, tempo řeči nebo přízvuk. Stejně tak i jeden konkrétní řečník může v různých situacích mít odlišně znějící hlas.

Dalším faktorem, který komplikuje rozpoznávání řeči, je akustické pozadí nahrávek. Máme-li k dispozici například promluvu z jedoucího automobilu nebo telefonní nahrávku z rušného prostředí, je rozpoznávání vyřčených slov výrazně obtížnější.

Úspěšnost rozpoznávání řeči je rovněž významně ovlivněná typem zpracovávané promluvy. Máme-li rozpoznávat pouze izolovaná slova, je úkol mnohem jednodušší, než pokud se jedná o diskrétní diktát nebo dokonce souvislou řeč.

Při rozpoznávání řeči jsou důležitá především dvě kritéria: úspěšnost rozpoznávání a vyžadovaný čas. V mnoha situacích je nutné provádět rozpoznávání řeči v reálném čase. V takovém případě rychlost výpočtu hraje velmi významnou roli. V případě složitějších úloh, zejména pokud se jedná o souvislou řečovou promluvu s velkým slovníkem, je někdy nutné zvolit vhodný kompromis mezi rychlostí a přesností rozpoznávání řeči.

## 1.1 Cíle diplomové práce

Cílem této práce je implementace vybraných metod automatického rozpoznávání řeči a jejich optimalizace s ohledem na co nejnižší čas výpočtu. V průběhu zpracování práce byl tento cíl rozšířen o analýzu nejen rychlosti, ale i úspěšnosti automatického rozpoznávání řeči.

# 2 Rozpoznávání řeči

Metody rozpoznávání řeči lze obecně rozdělit do dvou kategorií: metody založené na principu porovnávání se vzory, které využívají dynamické borcení času (DTW, Dynamic Time Warping), a statistické metody, které modelují slova nebo subslovní jednotky pomocí skrytých Markovových modelů (HMM, Hidden Markov Models).

V této práci je využíván statistický přístup k rozpoznávání.

## 2.1 Statistický přístup k rozpoznávání řeči

Princip statistických metod je následující: Posloupnost příznakových vektorů daného řečového signálu tvoří akustickou informaci $O = \{o_1 o_2 \dots_T\}$ a $W = \{w_1 w_2 \dots w_N\}$ je posloupnost $N$ slov. Cílem rozpoznávání je pak nalézt nejpravděpodobnější posloupnost slov pro danou akustickou informaci $O$, tedy posloupnost $\hat{W}$, která maximalizuje podmíněnou pravděpodobnost $P(W|O)$.

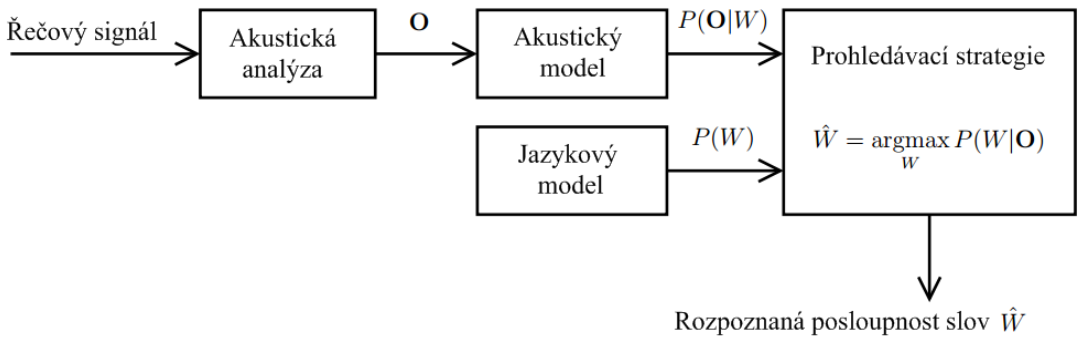
Tento požadavek lze dále upravit, využijeme-li Bayesova pravidla a zanedbáme-li pravděpodobnost $P(O)$, která nezávisí na $W$:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W)P(O|W) \tag{2.1}$$

Podmíněná pravděpodobnost $P(O|W)$ představuje informaci o akustickém modelu, zatímco $P(W)$ nese informaci o jazykovém modelu. Samotné rozpoznávání řeči lze pak rozdělit na čtyři základní úlohy:

1. Parametrizace řečového signálu - určení posloupnosti příznakových vektorů $O$.
2. Akustické modelování - vytvoření modelu pro výpočet podmíněné pravděpodobnosti $P(O|W)$
3. Jazykové modelování - vytvoření modelu pro výpočet pravděpodobnosti $P(W)$
4. Dekódování - určení nejpravděpodobnější posloupnosti slov $W$

Jednotlivé úlohy jsou blíže popsány v následujících kapitolách.




Obrázek 2.1: Blokové schéma systému rozpoznávání řeči [8]

# 3 Parametrizace řečového signálu

## 3.1 Význam parametrizace

Parametrizace řečového signálu představuje transformaci řečové nahrávky na soubor příznakových vektorů, charakterizujících hlavní vlastnosti signálu a s jejichž pomocí lze provést rozpoznání řeči.

Jedním z hlavních důvodů parametrizace je odstranění nadbytečné informace. Snažíme se získat příznaky, které co nejlépe reprezentují daný signál při relativně malé velikosti výsledných dat.

Při parametrizaci je rovněž nutné co nejlépe eliminovat zkreslení šumem na pozadí a další okolnosti, které mohou způsobit rozdíl mezi promluvami stejných slov. Výsledné vektory by pak měly být dostatečně „podobné“, aby bylo možné určit, že se jedná o stejná slova.

## 3.2 Metody krátkodobé analýzy

Metody krátkodobé analýzy vycházejí z předpokladu, že vlastnosti řečového signálu se v průběhu času mění pomalu. Toto plyne ze způsobu fungování hlasového traktu člověka. Proto lze rozdělovat řečový signál na krátké úseky a ty zpracovávat jako oddělené krátké zvuky. Tyto mikrosegmenty mají obvykle délku 10 ms. Vstupem bývají většinou data získaná digitalizací signálu, výsledkem analýzy jsou pak soubory čísel, popisující jednotlivé mikrosegmenty. [8]

Mezi nejpoužívanější metody patří metoda melovských kepstrálních koeficientů (MFCC) a perceptivní lineární prediktivní analýza (PLP).

## 3.3 Metoda melovských kepstrálních koeficientů

### 3.3.1 Preemfáze

Preemfáze se provádí před vlastním zpracováním řečového signálu. V mluvené řeči dochází k poklesu amplitud spektrálních složek řečového signálu na vyšších frekvencích. Toto se proto při preemfázi kompenzuje naopak zdůrazňováním těchto amplitud, a to dvěma způsoby:

a) analogovým filtrem

b) číslicovým filtrem podle vzorce:

$$y(n) = x(n) - ax(n-1), \tag{3.1}$$

kde $x(n)$ je n-tý vstupní vzorek, $y(n)$ n-tý výstup filtru a parametr $a$ je obvykle hodnota z rozmezí 0,9 až 1. [8]

### 3.3.2 Hammingovo okénko

Metody krátkodobé analýzy lze obvykle vyjádřit vztahem:

$$Q_n = \sum_{k=-\infty}^{\infty} \tau(s(k))w(n-k), \tag{3.2}$$

kde $Q_n$ je krátkodobá charakteristika, $s(k)$ vzorek akustického signálu, $\tau(.)$ transformační funkce a $w(n)$ je tzv. okénko - váhová posloupnost, kterou se váží vzorky s(k). Okénko slouží k výběru příslušných vzorků signálu a přidělení jim určité váhy při zpracování signálu. Zatímco pravoúhlé okénko přiřazuje všem vybraným vzorkům stejnou váhu, Hammingovo okénko slouží k potlačení vzorků na okrajích čímž se odstraní nespojitosti a vyhladí průběh spektra.

Pro Hammingovo okénko platí vztah:

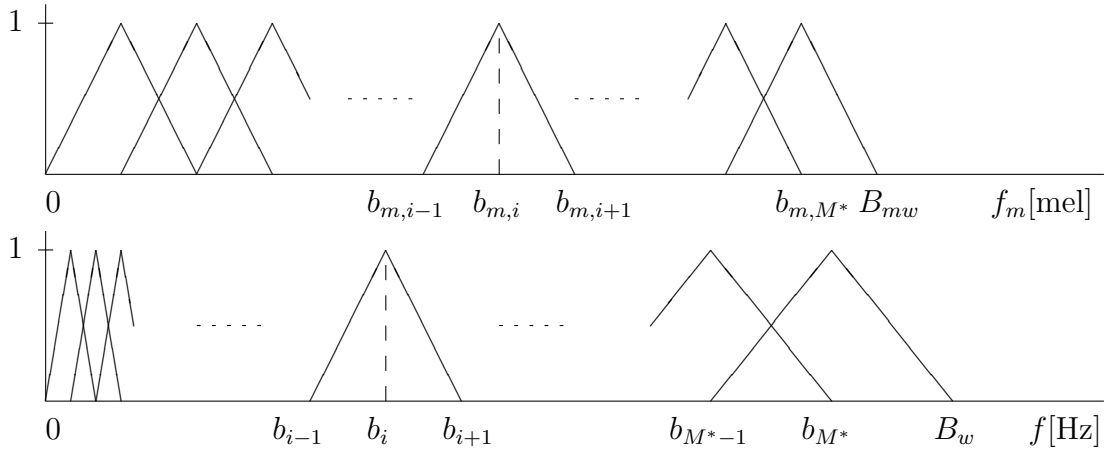
$$w(n) = \begin{cases} 0,54 - 0,46\cos(2\pi n/(L-1)) & \text{pro} \quad 0 \leq n \leq L-1, \\ 0 & \text{pro} \quad \text{ostatní } n, \end{cases} \tag{3.3}$$

kde $L$ je počet vzorků vybraných okénkem. Okénko se aplikuje na mikrosegmenty, které obvykle představují na sebe navazující úseky řečových vzorků o délce nejčastěji 10ms.

### 3.3.3 Popis metody MFCC

Metoda melovských kepstrálních koeficientů (Mel Frequency Cepstral Coefficients) vychází ze způsobu jakým řeč vnímá člověk. Změny ve výšce zvuku nejsou člověkem vnímány lineárně, ale spíše logaritmicky, a vyskytují se tzv. kritická pásma slyšení, představující frekvenční oblasti, kde dochází k maskování zvuku.

MFCC slouží zejména ke kompenzaci nelineárního vnímání frekvencí. Využívá se banka trojúhelníkových pásmových filtrů s lineárním rozložením frekvencí v melovské frekvenční ose, viz obrázek 3.1.




Obrázek 3.1: Banka trojúhelníkových filtrů v melovské a původní škále

Transformace mezi standardní a melovskou frekvenční osou je definovaná vztahem:

$$f_m = 2595 \log_{10}(1 + \frac{f}{700}), \tag{3.4}$$

kde $f$[Hz] je frekvence v lineární škále a $f_m$[mel] je odpovídající frekvence v nelineární melovské škále.[8]

Počet pásem $M^*$ je volen v závislosti na umístění a počtu kritických pásem, velikosti vzorkovací frekvence $F_v$[Hz] a celkové šířce přenášeného pásma $B_w$[Hz], resp. $B_{mw}$[mel].

Zpracování řečového signálu probíhá v několika krocích. Nejprve je provedena preemfáze a poté je na mikrosegmenty o délce zpravidla 10-30 ms aplikováno Hammingovo okénko. Následně je pomocí Fourierovy transformace proveden převod z časové do frekvenční oblasti, po čemž následuje výpočet amplitudového, případně výkonového spektra ($|S(f)|$, $|S(f)|^2$) analyzovaného signálu.

Nejdůležitějším krokem pak je melovská filtrace. Jsou aplikovány trojúhelníkové pásmové filtry, popsané vztahem:

$$u(f,i) = \begin{cases} \dfrac{f - b_{i-1}}{b_i - b_{i-1}} & \text{pro} \quad b_{i-1} \leq f < b_i, \\ \dfrac{f - b_{i+1}}{b_i - b_{i+1}} & \text{pro} \quad b_i \leq f < b_{i+1}, \\ 0 & \text{jinak}, \end{cases} \tag{3.5}$$

kde $u(f,i)$ je hodnota filtru a $b_{m,i}$ jsou střední frekvence jednotlivých filtrů.

Pro tyto střední frekvence platí v melovské škále vztah

$$b_{m,i} = b_{m,i-1} + \Delta_m, \qquad \text{kde} \quad b_{m,0} = 0, \quad i = 1, 2, ..., M^*, \quad \Delta_m = \frac{B_{mw}}{(M^* + 1)}. \tag{3.6}$$

Předposledním krokem je výpočet logaritmu výstupů $y_m(i)$ jednotlivých filtrů. Tímto se získá příznak, který mimo jiné vhodně omezí dynamiku signálu. Nakonec se provede zpětná diskrétní Fourierova transformace (IDFT). Jelikož výkonové spektrum je reálné a symetrické, lze IDFT redukovat na diskrétní kosinovou transformaci (DCT).

$$c_m(j) = \sum_{i=1}^{M^*} \log\, y_m(i)\, \cos\left[\frac{\pi j}{M^*}\,(i - 0,5)\right], \qquad j = 0, 1, \ldots, M, \tag{3.7}$$

kde $\{c_m(j)\}_{j=1}^{M}$ jsou výsledné melovské kepstrální koeficienty a jejich počet $M$ obvykle bývá volen podstatně menší než počet pásem filtru $M^*$.

### 3.3.4 Delta a delta-delta koeficienty

Dynamické koeficienty delta $\Delta c_m$ a delta-delta (akcelerační) $\Delta^2 c_m$ vyjadřují dynamiku časové změny vektorů příznaků. Pro jednotlivé mikrosegmenty se určují lineární regresí

$$[\Delta^{\Theta} c(j)]_n = \frac{\sum_{\kappa=0}^{\Theta} \kappa\{[c(j)]_{n+\kappa} - [c(j)]_{n-\kappa}\}}{2\sum_{\kappa=1}^{\Theta} \kappa^2}, \tag{3.8}$$

kde $[\Delta^{\Theta} c(j)]_n$ jsou delta koeficienty pro $n$-tý mikrosegment a $\Theta$ je odpovídající řád regrese.

Výsledný vektor příznaků pro každý mikrosegment je pak složen z melovských kepstrálních koeficientů, delta a delta-delta koeficientů.

## 3.4 Perceptivní lineární prediktivní analýza

Perceptivní lineární prediktivní analýza (PLP) představuje druhou z často používaných metod parametrizace řečového signálu. Podobně jako metoda MFCC bere v úvahu způsob vnímání různých frekvenčních složek signálu člověkem. Kombinuje přitom tři složky z psychofyziky slyšení: kritické pásmo spektrální citlivosti, křivky stejné hlasitosti a vztah vyjadřující závislost mezi intenzitou zvuku a jeho vnímanou hlasitostí. Sluchové spektrum je pak aproximováno autoregresivním celopólovým modelem.

PLP analýza sestává z následujících kroků [4]:

- Spektrální analýza

  Nejprve je provedena segmentace řečového signálu a každý segment je vážen Hammingovým okénkem (3.3.2). Poté je vypočteno výkonové spektrum signálu:

  $$P(\omega) = Re[S(\omega)]^2 + Im[S(\omega)]^2 \tag{3.9}$$

- Kritická pásma spektrální citlivosti

  Logaritmické vnímání změn ve výšce zvuku člověkem je modelováno pomocí nelineární transformace původní osy frekvencí $\omega$ [rad/s] na osu frekvencí $\Omega(\omega)$ měřenou v jednotce bark, podle vztahu

  $$\Omega(\omega) = 6 \ln\left(\frac{\omega}{1200\pi} + \sqrt{\left(\frac{\omega}{1200\pi}\right)^2 + 1}\right), \tag{3.10}$$

kde $\omega = 2\pi f$ [rad/s] a $\Omega(\omega)$ [bark]. Kritická pásma slyšení jsou pak simulována pomocí maskujících křivek, konstruovaných podle vztahů

$$\Psi(\Omega) = \begin{cases} 0 & \text{pro } \Omega < -1,3 \\ 10^{2,5(\Omega+0,5)} & \text{pro } -1,3 \leq \Omega \leq -0,5 \\ 1 & \text{pro } -05 < \Omega < 0,5 \\ 10^{(0,5-\Omega)} & \text{pro } 0,5 \leq \Omega \leq 2,5 \\ 0 & \text{pro } \Omega \geq 2,5 \end{cases} \tag{3.11}$$

Následně je provedena konvoluce funkce $\Psi(\Omega)$ s výkonovým spektrem $P(\Omega)$

$$\Theta(\Omega_i) = \sum_{\Omega=-1,3}^{2,5} P(\Omega - \Omega_i)\Psi(\Omega) \tag{3.12}$$

- Přizpůsobení kritických pásmových filtrů křivkám stejné hlasitosti

  Dalším krokem PLP analýzy je preemfáze vzorků $\Theta[\Omega(\omega)]$ křivkou stejné hlasitosti

  $$\Xi[\Omega(\omega)] = E(\omega)\Theta[\Omega(\omega)] \tag{3.13}$$

  Křivky stejné hlasitosti $E(\omega)$ aproximují rozdílnou citlivost lidského sluchu pro různé frekvence zvuku. Jejich přesná podoba závisí na zvolené hladině hlasitosti.

- Uplatnění vztahu vyjadřujícího závislost mezi intenzitou zvuku a vnímanou hlasitostí

  Pro simulaci nelineárního vztahu mezi intenzitou zvuku a vnímanou hlasitostí je na hodnoty $\Xi(\Omega)$ provedena operace třetí odmocniny. Tato operace zároveň snižuje amplitudovou proměnlivost kritických pásmových filtrů, což umožňuje provést následné celopólové modelování s relativně nízkým řádem modelu.

  $$\Phi(\Omega) = \sqrt[3]{\Xi(\Omega)} \tag{3.14}$$

- Aproximace spektrem celopólového modelu

  Posledním stádiem PLP analýzy je aproximace $\Phi(\Omega)$ spektrem celopólového modelu.

# 4 Akustické a jazykové modelování

## 4.1 Akustické modelování

Pro rozpoznávání řeči se velmi často používají skryté Markovovy modely (Hidden Markov Model, HMM). Tento přístup vychází z myšlenky, že při vytváření řeči člověkem je hlasové ústrojí v průběhu krátkého časového intervalu ve stejném, neměnném stavu. Přitom je jím vytvářen zvukový signál, který lze popsat pomocí vektoru příznaků.

Posloupnost příznakových vektorů všech mikrosegmentů daného řečového signálu tvoří akustickou informaci $O$. Úkolem akustického modelu potom je poskytnout co nejlepší odhad podmíněné pravděpodobnosti $P(O|W)$ pro libovolnou posloupnost slov $W$.

### 4.1.1 Struktura skrytých Markovových modelů

Skrytý Markovův model je model stochastického procesu s neznámými stavy generujícími posloupnost pozorování. Model v každém časovém kroku mění svůj stav na základě pravděpodobností $a_{ij}$ přechodu ze stavu $s_i$ do stavu $s_j$. Jednotlivé stavy generují vektory pozorování $o_t$ podle rozdělení výstupní pravděpodobnosti $b_j(o_t)$.

Pro modelování řeči jsou používány zejména tzv. levo-pravé Markovovy modely. U těchto modelů proces začíná v počátečním stavu modelu, a v každém dalším časovém okamžiku model setrvá ve stejném stavu nebo přejde do stavu s vyšším indexem. Po ukončení procesu se model nachází v koncovém stavu. Příklad levo-pravého skrytého Markovova modelu je ukázán na obrázku 4.1. První a poslední stav znázorněného modelu jsou tzv. neemitující stavy - stavy, které negenerují žádná pozorování. Zbylé stavy jsou emitující.

Funkce $b_j$ popisuje rozdělení pravděpodobnosti pozorování $o_t$ produkovaného ve stavu $s_j$ v čase $t$. Pro pozorování, nabývající spojitých hodnot, představuje funkce $b_j(o_t)$ hustotu pravděpodobnosti jevu, že stav $s_j$ v čase $t$ generuje pozorování $o_t$.

Jedním z nejčastěji využívaných rozdělení pravděpodobnosti je spojité rozdělení se směsí Gaussových hustotních funkcí. Výstupní hustota pravděpodobnosti je poté tvořena váženým součtem určitého počtu normálních hustot pravděpodobnosti daných středními hodnotami

a kovariančními maticemi.

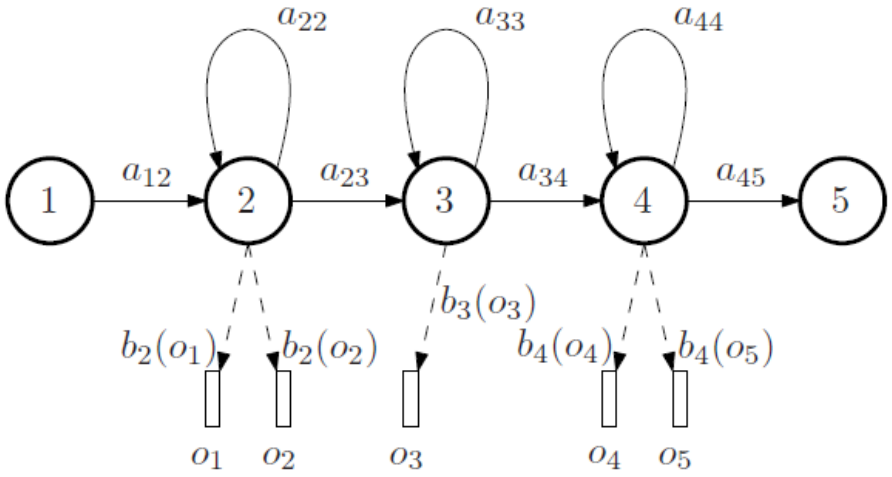
Hustotní funkce $b_j$ je při použití směsi normálních funkcí vyjádřena následujícím vztahem:

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} N(o_t; \mu_{jm}, C_{jm}), \tag{4.1}$$

$$N(o_t; \mu_{jm}, C_{jm}) = \frac{1}{\sqrt{(2\pi)^n |C_{jm}|}} \exp(-\frac{1}{2}(o_t - \mu_{jm})^T C_{jm}^{-1}(o_t - \mu_{jm})), \tag{4.2}$$

kde $M$ je počet složek hustotní směsi, $n$ je dimenze vektoru pozorování $o_t$, $c_{jm}$ představuje váhu $m$-té složky $j$-tého stavu, $N(o_t; \mu_{jm}, C_{jm})$ je normální rozdělení se střední hodnotou $\mu_{jm}$ a kovarianční maticí $C_{jm}$.

Hustotní funkce $b_j$ je obecně tvořena směsí $M$ normálních rozdělení, s plnou kovarianční maticí. Model použitý v rámci diplomové práce však pro zjednodušení obsahoval pouze jednoduché normální rozdělení ($M = 1$) a kovarianční matice byla diagonální.




Obrázek 4.1: Skrytý Markovův model s pěti stavy (převzato z [1])

### 4.1.2 Pravděpodobnost generování promluvy modelem

Úkolem akustického modelu potom je poskytnout co nejlepší odhad podmíněné pravděpodobnosti $P(O|W)$ pro libovolnou posloupnost slov $W$ a posloupnost vektorů příznaků

$O$. Jestliže posloupnost $W$ je modelována odpovídajícím skrytým Markovovým modelem $\lambda$, pak podmíněnou pravděpodobnost $P(O|W)$ lze nahradit výpočtem podmíněné pravděpodobnosti $P(O|\lambda)$. Jelikož posloupnost stavů $S = \{s(0), s(1), \ldots, s(T+1)\}$ je skrytá, podmíněnou pravděpodobnost $P(O|\lambda)$ je nutné vypočítat jako součet pravděpodobností pro všechny možné posloupnosti stavů $S$.

Platí tedy

$$P(O|\lambda) = \sum_S P(O, S|\lambda) = \sum_S P(O|S, \lambda)P(S|\lambda) = \sum_S a_{s(0)s(1)} \prod_{t=1}^{T} b_{s(t)}(o_t) a_{s(t)s(t+1)}. \quad (4.3)$$

Vzhledem k obrovskému množství vyžadovaných operací přímý výpočet $P(O|\lambda)$ podle uvedeného vztahu není používán. Pro efektivnější výpočet proto slouží algoritmus forward-backward, popsaný v části 4.1.3, který značně snižuje vyžadovaný počet operací násobení.

Další možnou alternativou je rovněž aproximace výpočtu podmíněné pravděpodobnosti $P(O|\lambda)$ pravděpodobností $P_S(O|\lambda)$ nejpravděpodobnější posloupnosti stavů, kterou projde posloupnost $O$ modelem $\lambda$. Tuto posloupnost a pravděpodobnost $P_S(O|\lambda)$ lze určit pomocí Viterbiova algoritmu, který je popsán v části 4.1.4.

### 4.1.3 Algoritmus forward-backward

Algoritmus forward-backward je iterační algoritmus sloužící k výpočtu podmíněné pravděpodobnosti $P(O|\lambda)$ během dvou průchodů.

Během průchodu odpředu počítáme sdruženou pravděpodobnost $\alpha_j(t)$ pozorování prvních $t$ řečových vektorů $\{o_1 o_2 \ldots o_t\}$ a jevu, že proces se nachází v čase $t$ ve stavu $s_j$, za podmínky modelu $\lambda$.

$$\alpha_j(t) = P(o_1 o_2 \ldots o_t, s(t) = s_j|\lambda) \quad (4.4)$$

Během průchodu odzadu počítáme podmíněnou pravděpodobnost $\beta_j(t)$ pozorování posledních $T - t$ řečových vektorů $\{o_{t+1} o_{t+2} \ldots o_T\}$ za podmínky, že model $\lambda$ se nachází v čase $t$ ve stavu $s_j$.

$$\beta_j(t) = P(o_{t+1} o_{t+2} \ldots o_T|s(t) = s_j, \lambda) \quad (4.5)$$

Výpočet probíhá podle následujících vztahů:

Výpočet odpředu:

$$\alpha_1(1) = 1 \tag{4.6}$$

$$\alpha_j(1) = a_{1j} b(o_t) \qquad \text{pro } 1 < j < N \tag{4.7}$$

$$\alpha_j(t) = \left[ \sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij} \right] bj(o_t) \qquad \text{pro } 1 < j < N, t = 2, 3, \ldots, T \tag{4.8}$$

Výpočet odzadu:

$$\beta_i(T) = a_{iN} \qquad \text{pro } 1 < i < N \tag{4.9}$$

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} bj(o_{t+1}) \beta_j(t+1) \qquad \text{pro } 1 < i < N, t = T-1, \ldots, 1 \tag{4.10}$$

Pro výslednou pravděpodobnost poté platí

$$P(O|\lambda) = \sum_{i=2}^{N-1} \alpha_i(t) \beta_i(t). \tag{4.11}$$

Jelikož algoritmus vede k operacím s velmi malými hodnotami, u kterých hrozí podtečení, ve výpočtu jsou obvykle použity logaritmy pravděpodobností a operace násobení jsou poté nahrazeny součty. Rovněž lze v průběhu výpočtu provádět normalizaci získaných hodnot, což zejména v případě dlouhých nahrávek také zvyšuje numerickou stabilitu výpočtu.

Kromě požadované podmíněné pravděpodobnosti $P(O|\lambda)$ lze pomocí algoritmu forward-backward získat rovněž nejpravděpodobnější posloupnost stavů, kterou posloupnost $O$ projde modelem $\lambda$. Tu získáme pomocí matice $\Gamma$, jejíž prvky $\gamma_i(t)$ představují pravděpodobnosti, že model se v čase $t$ nachází ve stavu $s_i$, za podmínky modelu $\lambda$ a generování posloupnosti pozorování $O$, a platí pro ně vztah

$$\gamma_i(t) = P(s(t) = s_i|O, \lambda) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{i=2}^{N-1} \alpha_i(t)\beta_i(t)}. \tag{4.12}$$

Maximalizací hodnot $\gamma_i(t)$ přes všechny stavy $s_i$ poté nalezneme nejpravděpodobnější posloupnost stavů.

### 4.1.4 Viterbiův algoritmus

Viterbiův algoritmus slouží k nalezení nejpravděpodobnější posloupnosti skrytých stavů. Výpočet se skládá ze dvou částí, dopředného a zpětného běhu.

Činnost algoritmu lze přirovnat k hledání cesty s nejlepší cenou v ohodnoceném grafu. Během dopředného běhu algoritmus sleduje v každém časovém kroku $t$ nejpravděpodobnější posloupnost stavů, kterou se lze během prvních $t$ časových okamžiků dostat z počátečního stavu do každého ze stavů. Přitom je pamatována nejvyšší pravděpodobnost, jaké bylo v aktuálním čase v každém stavu dosaženo, a dále je pro každou dvojici stavů a časů pamatován stav modelu v předchozím časovém okamžiku. Tato informace je poté využita při zpětném trasování pro nalezení nejlepší cesty.

$$\varphi_j(t) = \max_{s(1),\ldots,s(t-1)} P(o_1 \ldots o_t, s(1), s(2), \ldots, s(t) = s_j|\lambda) \tag{4.13}$$

Algoritmus sestává z následujících kroků, kde $\varphi_j(t)$ značí pravděpodobnost maximálně pravděpodobné posloupnosti stavů $s(1), s(2), \ldots, s(t) = s_j$ v každém časovém kroku $t$ pro každý stav $s_j$ a proměnná $\psi_j(t)$ představuje informaci o tom, ze kterého stavu $s(t-1)$ se lze v čase $t$ dostat do $s_j$ s nejvyšší výslednou pravděpodobností:

1. Inicializace

$$\varphi_j(1) = a_{1j} b_j(o_1) \qquad \text{pro } j = 2, \ldots, N-1 \tag{4.14}$$

$$\psi_j(1) = 0 \tag{4.15}$$

2. Rekurze pro $t = 2, 3, \ldots, T$ a $j = 2, \ldots, N-1$

$$\varphi_j(t) = \left\{ \max_{i=2,\ldots,N-1} [\varphi_i(t-1) a_{ij}] \right\} b_j(o_t) \tag{4.16}$$

$$\psi_j(t) = \operatorname*{argmax}_{i=2,\ldots,N-1} [\varphi_i(t-1) a_{ij}] \tag{4.17}$$

3. Výsledná pravděpodobnost a index maximálně pravděpodobného stavu v čase $T$

$$P_S(O|\lambda) = \max_{i=2,\ldots,N-1} [\varphi_i(T) a_{iN}] \tag{4.18}$$

$$i_T^* = \underset{i=2,\ldots,N-1}{\operatorname{argmax}} \left[\varphi_i(T) a_{iN}\right] \tag{4.19}$$

Posloupnost stavů lze určit zpětným trasováním podle vztahu

$$i_t^* = \psi_{i_{t+1}^*}(t+1). \tag{4.20}$$

Vzhledem k riziku podtečení hodnot jsou podobně jako u algoritmu forward-backward při výpočtu obvykle používány logaritmy pravděpodobností, případně v kombinaci s průběžnou normalizací získávaných logaritmů pravděpodobností.

Trénování akustického modelu bylo prováděno pomocí algoritmu forward-backward, Viterbiův algoritmus byl však přesto v rámci práce využit ve dvou podobách. Ve tvaru, který je popsán zde, byl tento algoritmus aplikován při vyhodnocování výsledků za účelem zjištění konkrétní posloupnosti stavů, odpovídající referenčnímu přepisu promluv z testovací množiny. Vedle tohoto Viterbiův algoritmus po určitých změnách sloužil rovněž ve fázi dekódování pro nalezení výsledné posloupnosti slov vyřčených v nahrávkách.

### 4.1.5 Trénování parametrů skrytého Markovova modelu

Parametry skrytého Markovova modelu získáme jejich natrénováním na množině trénovacích dat. K tomuto účelu se nejčastěji používá metoda maximální věrohodnosti (Maximum Likelihood, ML).

Cílem je nalézt takové parametry modelu $\lambda$, které pro daný soubor $E$ trénovacích promluv $\{O^e\}_{e=1}^E$ maximalizují věrohodnostní funkci $F(O^1, O^2, \ldots, O^E|\lambda)$, definovanou

$$F(O^1, O^2, \ldots, O^E|\lambda) = \prod_{e=1}^{E} P(O^e|\lambda). \tag{4.21}$$

Často se pracuje s logaritmem věrohodnostní funkce, hledáme pak tedy parametry $\hat{\lambda}$, splňující vztah

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} \sum_{e=1}^{E} \log P(O^e|\lambda). \tag{4.22}$$

Pro nalezení parametrů $\hat{\lambda}$ bývá využíván Baum-Welchův reestimační algoritmus, který

představuje variantu algoritmu EM (Expectation-Maximization, popis a odvození lze nalézt například v [8]), určenou speciálně pro odhadování parametrů skrytých Markovových modelů. Jedná se o iterativní algoritmus, upravující parametry HMM tak, aby co nejlépe odpovídaly pozorované posloupnosti příznakových vektorů $O$.

Trénovací cyklus sestává z těchto kroků:

1. Zvolíme počáteční odhady parametrů $\mu_{jm}$, $C_{jm}$

2. Pomocí algoritmu forward-backward vypočítáme pro každou trénovací promluvu $e$ hodnoty $P(O^e|\lambda)$, $\gamma_j^e(t)$ a $\gamma_{jm}^e(t)$

$$P(O^e|\lambda) = \sum_{j=2}^{N-1} \alpha_j^e(t)\beta_j^e(t) \tag{4.23}$$

$$\gamma_j^e(t) = \frac{\alpha_j^e(t)\beta_j^e(t)}{P(O^e|\lambda)} \tag{4.24}$$

$$\gamma_{jm}^e(t) = \begin{cases} \dfrac{1}{P(O^e|\lambda)}\, a_{1j}c_{jm}b_{jm}(o_t^e)\beta_j^e(t) & \text{pro } t = 1, \\ \dfrac{1}{P(O^e|\lambda)} \displaystyle\sum_{i=2}^{N-1} \alpha_i^e(t-1)a_{ij}c_{jm}b_{jm}(o_t^e)\beta_j^e(t) & \text{pro } t \geq 1 \end{cases} \tag{4.25}$$

3. Na základě hodnot získaných v kroku 2 vypočítáme hodnoty parametrů

$$\bar{\mu}_{jm} = \frac{\displaystyle\sum_{e=1}^{E}\sum_{t=1}^{T_e} \gamma_{jm}^e(t) o_t^e}{\displaystyle\sum_{e=1}^{E}\sum_{t=1}^{T_e} \gamma_{jm}^e(t)} \tag{4.26}$$

$$\bar{C}_{jm} = \frac{\displaystyle\sum_{e=1}^{E}\sum_{t=1}^{T_e} \gamma_{jm}^e(t)(o_t^e - \bar{\mu}_{jm})(o_t^e - \bar{\mu}_{jm})^T}{\displaystyle\sum_{e=1}^{E}\sum_{t=1}^{T_e} \gamma_{jm}^e(t)} \tag{4.27}$$

4. Pokud se hodnoty parametrů změnily, provedeme návrat do kroku 2. V opačném případě trénování končí.

## 4.2 Jazykové modelování

Jazykový model slouží pro odhadování pravděpodobnosti $P(W)$ pro libovolnou posloupnost slov $W$. Vychází při tom ze zákonitostí daného jazyka, a to zejména jeho slovníku a pravidel řetězení slov do vět.

Stochastické jazykové modely určují pravděpodobnost $P(W)$ podle vztahu

$$P(W) = P(w_1 w_2 \dots w_N) = \prod_{n=1}^{N} P(w_n | w_1 \dots w_{n-1}). \tag{4.28}$$

Mezi hlavní typy stochastických jazykových modelů patří $n$-gramové modely. Ty vyjadřují pravděpodobnost výskytu každého slova v promluvě v závislosti na pouze $n-1$ předchozích slovech. Nejčastějšími variantami jsou trigramové ($n = 3$), bigramové ($n = 2$) a unigramové ($n = 1$) modely. Zerogramové modely ($n = 0$) představují speciální případ unigramových, u kterých pravděpodobnosti všech slov jsou shodné [3]:

trigramový model: $P(w_n|w_1 \dots w_{n-1}) = P(w_n|w_{n-2}, w_{n-1})$
bigramový model: $P(w_n|w_1 \dots w_{n-1}) = P(w_n|w_{n-1})$
unigramový model: $P(w_n|w_1 \dots w_{n-1}) = P(w_n)$
zerogramový model: $P(w_n|w_1 \dots w_{n-1}) = \frac{1}{V}$

Hodnota $V$ zde představuje celkový počet slov slovníku.

Konstrukce vhodného jazykového modelu může být velmi obtížnou záležitostí, zejména pokud je rozpoznávána souvislá řeč, pro potřeby úlohy zpracovávané v rámci diplomové práce však postačoval velmi jednoduchý model. Ten je schematicky znázorněn na obr. 7.1.

Pro rozpoznávání řeči je rovněž nutné nastavit parametr váhy jazykového modelu. Pokud má jazykový model příliš velkou váhu, výsledný dekodér se řídí více očekáváními založenými na předchozích rozpoznaných slovech, než skutečně vyslovenou promluvou. Naopak pokud má výrazně vyšší váhu akustický model, dekodér zvolí slovo, které tento model určí k vyslovenému jako nejpodobnější, bez ohledu na jeho vhodnost v daném kontextu. [10]

# 5 Dekódování

Posledním stádiem úlohy rozpoznání řeči, po vytvoření akustického a jazykového modelu, je dekódování, neboli určení nejpravděpodobnější posloupnosti slov $W$ odpovídající posloupnosti pozorování $O$. K tomuto účelu se využijí pravděpodobnosti $P(O|W)$ a $P(W)$. Podmíněnou pravděpodobnost $P(O|W)$ lze získat pomocí akustického modelu, pravděpodobnost $P(W)$ pak pomocí modelu jazykového, pokud je použit.

Účelem dekodéru pak je pro posloupnost příznakových vektorů $O = \{o_1 o_2 \dots o_T\}$ promluvy nalézt takovou posloupnost slov, pro kterou součin těchto dvou pravděpodobností nabývá maximální hodnoty. Hledáme tedy posloupnost slov $\hat{W} = w_1 w_2 \dots w_M$, pro kterou platí

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W)P(O|W). \tag{5.1}$$

## 5.1 Definice úlohy dekódování

Úlohu dekódování lze definovat na základě dvou různých kritérií, a to kritéria MAP (maximální aposteriorní pravděpodobnosti) a podle Viterbiova kritéria. [8]

Dle kritéria MAP je úloha dekódování definovaná jako hledání posloupnosti slov $\hat{W}$, která vyhovuje vztahu

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{ P(W) \sum_{S \in \Phi_W} P(O|W)P(S|W) \right\}, \tag{5.2}$$

kde $\Phi_W$ značí množinu všech stavových posloupností $S$ reprezentujících konkrétní posloupnost slov $W$.

Viterbiovo kritérium pak předchozí vztah zjednodušuje omezením se pouze na nejpravděpodobnější posloupnost stavů, tedy

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{ P(W) \max_{S} P(O|W)P(S|W) \right\}. \tag{5.3}$$

## 5.2 Časově synchronní Viterbiovo prohledávání

Jednou z možností dekódování je časově synchronní prohledávání stavového prostoru pomocí Viterbiova algoritmu.

Pro dekódování je využívána tzv. rozpoznávací mřížka (recognition trellis). Ta představuje graf, jehož uzly jsou tvořeny jednotlivými stavy v konkrétních časových okamžicích. Hrany pak představují přechod modelu z jednoho stavu do dalšího v následujícím čase. Pro každou možnou posloupnost stavů existuje právě jedna cesta grafem.[2]

Jestliže jednotlivé hrany grafu jsou ohodnoceny zápornými logaritmy pravděpodobností, poté úloha dekódování ve své podstatě představuje pouhý problém hledání cesty s nejmenší cenou v ohodnoceném orientovaném grafu.

1. Inicializace

$$\Phi_j(1) = \begin{cases} -\log b_j(o_1) & \text{pro } j = 2 \\ -\log(0) = \infty & \text{pro } j = 3, \ldots, N-1 \end{cases} \tag{5.4}$$

2. Rekurze pro $t = 2, \ldots, T$, $j = 2, \ldots, N-1$

$$\Phi_j(t) = \min_{i=2}^{N-1}(\Phi_i(t-1) - \log a_{ij}) - \log b_j(o_t) \tag{5.5}$$

$$\Psi_j(t) = \operatorname*{argmin}_{i=2}^{N-1}(\Phi_i(t-1) - \log a_{ij}) - \log b_j(o_t) \tag{5.6}$$

3. Výsledek

$$-\log P(O|w) = \Phi_{N-1}(T), \tag{5.7}$$

$$\Psi_{N-1}(T) = N - 1, \tag{5.8}$$

$T$ je počet vektorů pozorování, $N$ je počet stavů, $a_{ij}$ vyjadřuje pravděpodobnost přechodu ze stavu $s_i$ do stavu $s_j$, $b_j(o_t)$ je výstupní pravděpodobnost pozorování $o_t$ ve stavu $s_j$. Funkce $\Phi_j(t)$ představuje kumulativní ohodnocení uzlu $s_j(t)$ a funkce $\Psi_j(t)$ obsahuje index stavu předcházejícího stavu $s_j$ pro čas $t$. [8]

Nejpravděpodobnější posloupnost stavů získáme zpětným trasováním pomocí funkce $\Psi$.

Popsaný algoritmus slouží pro hledání cesty pro jedno slovo, nebo pro pevně danou posloupnost slov. Toho bylo využito při zjišťování „správné“ posloupnosti stavů s využitím

referenčního přepisu promluv. Pro rozpoznávání neznámé posloupnosti slov byl výpočet mírně upraven. Množina stavů zahrnovala stavy všech slov a hrany grafu byly rozšířeny o přechody ze stavu reprezentujícího ticho do prvního stavu každého slova a z posledních stavů slov opět do ticha. To umožňovalo rozpoznávání všech posloupností slov vyhovujících použitému jazykovému modelu (obr. 7.1).

# 6 Optimalizace úlohy rozpoznávání řeči z hlediska času výpočtu

Rozpoznávání řeči představuje z hlediska výpočetního času poměrně náročnou úlohu. Nejnáročnější část přitom představuje dekódování a zejména výpočet akustických pravděpodobností se významným dílem podílí na celkovém čase výpočtu.

## 6.1 Zrychlení výpočtu akustických pravděpodobností

### 6.1.1 Převod výpočtu na kombinaci dvou jednodušších operací

Výpočet akustických pravděpodobností představuje jednu z časově nejnáročnějších částí úlohy rozpoznávání řeči. Jestliže však má použitý HMM diagonální kovarianční matice, lze tento výpočet zjednodušit. Jeden z možných přístupů představuje výpočet akustické pravděpodobnosti pomocí kombinace dvou jednodušších a snáze optimalizovatelných operací, a to násobení matic a funkce *addLog*, definované

$$\text{addLog}(x_1, x_2) = \ln(e^{x_1} + e^{x_2}). \tag{6.1}$$

Postup odvození je následující [7]:

Akustická pravděpobnost $b_j(o_t)$ generování příznakového vektoru $o_t$ stavem $s_j$ je definovaná již dříve uvedeným vztahem

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sqrt{(2\pi)^n |C_{jm}|}} \exp(-\frac{1}{2}(o_t - \mu_{jm})^T C_{jm}^{-1}(o_t - \mu_{jm})), \tag{6.2}$$

kde $c_{jm}$ je váha $m$-té složky gaussovské směsi, $\mu_{jm}$ značí střední hodnotu, $C_{jm}$ kovarianční matici a $n$ je počet dimenzí příznakových vektorů.

Pro každou ze složek gaussovské směsi pak lze logaritmus pravděpodobnosti vypočítat

podle vztahu

$$\ln b_{jm}(o_t) = \ln c_{jm} - \frac{1}{2}\ln((2\pi)^n|C_{jm}|) - \frac{1}{2}\mu_{jm}^T C_{jm}^{-1}\mu_{jm} + \mu_{jm}^T C_{jm}^{-1} o_t - \frac{1}{2}o_t^T C_{jm}^{-1} o_t, \quad (6.3)$$

První tři členy nezávisejí na $o_t$, lze je tedy pro model vypočítat předem. Označíme je jako $h_{jm}$.

$$h_{jm} = \ln c_{jm} - \frac{1}{2}\ln((2\pi)^n|C_{jm}|) - \frac{1}{2}\mu_{jm}^T C_{jm}^{-1}\mu_{jm} \quad (6.4)$$

Pro další dva členy je zavedena substituce

$$U_{jm} = \mu_{jm}^T C_{jm}^{-1} \quad (6.5)$$

$$V_{jm} = \mathrm{diag}(-\frac{1}{2}C_{jm}^{-1}), \quad (6.6)$$

kde $C_{jm}$ je diagonální kovarianční matice, jejíž prvky na diagonále jsou $\{\sigma_{11}, \ldots, \sigma_{nn}\}$. Potom logaritmus pravděpodobnosti $b_j(o_t)$ lze vyjádřit jako

$$\ln b_{jm}(o_t) = h_{jm} + U_{jm}o_t + V_{jm}^T o_t^2, \quad (6.7)$$

což lze vypočítat jako skalární součin

$$A = (1, o_{t1}, o_{t2}, \ldots, o_{tn}, o_{t1}^2, o_{t2}^2, \ldots, o_{tn}^2) \quad (6.8)$$

$$B_{jm} = (h_{jm}, \mu_1\sigma_{11}^{-1}, \ldots, \mu_n\sigma_{nn}^{-1}, -\frac{1}{2}\sigma_{11}^{-1}, \ldots, -\frac{1}{2}\sigma_{nn}^{-1}), \quad (6.9)$$

kde $o_{tn}$ značí $n$-tý prvek příznakového vektoru $o_t$ a $\mu_n$ představuje $n$-tý prvek vektoru středních hodnot $\mu_{jm}$.

Celkovou logaritmickou akustickou pravděpodobnost poté lze vypočítat pomocí vztahu

$$\ln b_j(o_t) = \operatorname*{addLog}_{m=1}^{M}(A \cdot B_{jm}), \quad (6.10)$$

Výpočet logaritmu akustické pravděpodobnosti byl tedy takto převeden na kombinaci násobení matic a funkce *addLog*, definované rovnicí 6.1.

Pro zrychlení výpočtu násobení matic se nabízí využití specializovaných knihoven funkcí, optimalizovaných pro konkrétní typy procesorů, které poskytují výrazně rychlejší výpočet

než klasický způsob implementace této operace.

Pro zrychlení výpočtu funkce *addLog* pak byla použita aproximace Taylorovým rozvojem. Koeficienty tohoto rozvoje byly předpočítány pro některé hodnoty a uloženy v tabulce, ve které byly následně během vlastního výpočtu vyhledávány. Smyslem této tabulky je ušetřit čas v případě, že provádíme velké množství těchto operací, jelikož ji stačí vygenerovat pouze jednou.

### 6.1.2 Aproximace výpočtu funkce *addLog*

Vztah 6.1 lze převést do alternativního tvaru

$$\text{addLog}(x_1, x_2) = \ln(e^{x_1} + e^{x_2}) = \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|}). \tag{6.11}$$

Poté stačí zabývat se pouze výrazem $\ln(1 + e^x)$, kde $x = -|x_1 - x_2|$.

Aproximace sestává ze dvou základních kroků. Nejdříve se předpočítá prvních $N$ koeficientů (kde $N$ je požadovaný stupeň rozvoje) Taylorova rozvoje funkce $f(x) = \ln(1+e^x)$ v určitých bodech $x_0$ a tyto koeficienty se uloží do tabulky.

$$a_n(x_0) = \frac{f^{(n)}(x_0)}{n!} \tag{6.12}$$

Během samotného výpočtu se poté pro každou dvojici $x_1$ a $x_2$ vypočítá $x = -|x_1 - x_2|$ a pro něj se určí nejbližší hodnota, pro niž jsou koeficienty obsažené v tabulce. S jejich pomocí se poté vypočítá hodnota rozvoje v daném bodě a následně celý hledaný výsledek.

$$T(x) = \sum_{n=0}^{N} a_n(x_0)(x - x_0)^n \tag{6.13}$$

$$\hat{y} = \max(x_1, x_2) + T(x) \tag{6.14}$$

### 6.1.3 Volba parametrů

V rámci výběru konkrétních variant algoritmu bylo potřeba vhodně zvolit tři různé parametry:

- **Rozmezí bodů v tabulce:**
  Vyhledávací tabulka má konečné rozměry, proto se musí zvolit maximální rozdíl mezi $x_1$ a $x_2$, pro který jsou v ní předpočítané hodnoty. V případě většího rozdílu se počítá s touto hodnotou. Hranice musí být tedy stanovena tak, aby takto způsobená chyba byla menší než chyba vzniklá následným výpočtem. Na základě experimentů bylo zjištěno, že pro omezení vlivu na přesnost výpočtu je potřeba vytvořit tabulku pro hodnoty z intervalu až $< 0, 40 >$. V případě výpočtu ve floatové přesnosti však postačuje $< 0, 20 >$.

- **Stupeň rozvoje:**
  Stupeň Taylorova rozvoje má výrazný vliv na přesnost aproximace, avšak také na rychlost, jelikož každý přidaný člen rozvoje znamená několik matematických operací navíc pro každou dvojici $x_1$ a $x_2$. Proto je zde potřeba zvolit vhodný kompromis.

- **Délka tabulky:**
  Teoreticky vyšší počet bodů v tabulce zlepšuje přesnost výpočtu bez podstatného vlivu na jeho rychlost, toto však platí pouze pokud se celá tabulka vejde do vyrovnávací paměti. Proto pro dosažení nejlepších výsledků nestačí pouze použít nízký stupeň rozvoje s rozsáhlou tabulkou.

Zvoleny byly nakonec tři kombinace parametrů, lišící se časem výpočtu a přesností získaného výsledku:

1. float přesnost, Taylorův rozvoj 2. stupně, tabulka pro 512 hodnot z $< 0, 20 >$
2. double přesnost, Taylorův rozvoj 3. stupně, tabulka pro 2048 hodnot z $< 0, 40 >$
3. double přesnost, Taylorův rozvoj 5. stupně, tabulka pro 1024 hodnot z $< 0, 40 >$

Pro tyto tři varianty algoritmu byl následně dále výpočet optimalizován vzhledem k času, a to zejména s využitím instrukční sady SSE (Streaming SIMD Extensions). Ta umožňuje provádět některé operace s více hodnotami najednou (až 128 bitů, tzn. například 2 hodnoty typu double nebo 4 hodnoty float).

Jelikož obecně i po sobě jdoucí vstupní data odpovídají různým místům vyhledávací tabulky, všechny potřebné hodnoty se nejdříve kopírují do samostatného pole tak, aby SSE instrukce mohly pracovat se dvěma nebo čtyřmi (podle varianty) hodnotami najednou.

### 6.1.4 Výsledky

Pro posouzení přesnosti jednotlivých variant bylo jako vstup použito velké množství náhodně vygenerovaných hodnot s normálním rozdělením pravděpodobnosti $x_1, x_2 \sim N(0, 256)$. Samotná přesnost pak byla posuzována pomocí odmocniny ze střední kvadratické chyby (RMSD) oproti výsledku získanému přesnějším výpočtem pomocí Matlabu (dále označováno jako „chyba“).

Následující tabulka představuje změřené časy a přesnosti pro výpočet 50.000.000 hodnot zvolenými variantami aproximace a přesným výpočtem pomocí Matlabu. U aproximace časy odpovídají pouze samotnému výpočtu, bez tvorby tabulky, typových konverzí apod.

MALPS = Mega AddLog Per Second, tzn. spočtené hodnoty vyjádřené v milionech za vteřinu.

| Výpočet 50.000.000 hodnot | | | |
|---|---|---|---|
| varianta | chyba | čas | MALPS |
| přesný výpočet (double) | 0 | 1,35 s | 37,1 |
| přesný výpočet (float) | $4,5 \cdot 10^{-7}$ | 1,92 s | 26,1 |
| T. 2. stupně, float | $4,5 \cdot 10^{-7}$ | 0,32 s | 155,6 |
| T. 3. stupně, double | $2,6 \cdot 10^{-12}$ | 0,70 s | 71,4 |
| T. 5. stupně, double | $1,5 \cdot 10^{-15}$ | 0,78 s | 64,4 |

Tabulka 6.1: Srovnání časů a přesností výpočtu 50 milionů hodnot pro různé varianty

Tabulka ukazuje, že v závislosti na zvolené variantě bylo dosaženo přibližně 40-75% zrychlení výpočtu při zachování poměrně vysoké přesnosti.

Zrychlení operace addLog hraje velkou roli zejména u rozpoznávání v reálném čase. Při zpracování velkých akustických modelů jsou prováděny desítky milionů operací AddLog za sekundu, přesný výpočet tedy zabírá podstatnou část výpočetních schopností procesoru. Použitím aproximace se však tento požadavek sníží, a zbude tak více prostředků pro násobení matic a dekódování. Není poté třeba dělat kompromisy mezi rychlostí a přesností rozpoznávání, a tedy výsledný systém dosahuje nejlepších možných výsledků i v úlohách vyžadujících zpracování v reálném čase.

## 6.2 Zrychlení dekódování pomocí prořezávání

Výpočetní náročnost dekodéru lze rovněž snížit vhodným prořezáváním stavového prostoru během výpočtu. Prořezávání bylo prováděno konkrétně pro algoritmus forward-backward, avšak stejný princip lze aplikovat i na algoritmus Viterbiův. Prořezávání lze rovněž kromě dekódování využít i během trénování akustického modelu, avšak v praxi to není tolik obvyklé, jelikož se zde klade větší důraz na přesnost výpočtu.

Během výpočtu algoritmu forward-backward je počítáno s maticemi $N \times T$ hodnot $\alpha$ a $\beta$, kde $N$ je počet stavů a $T$ počet časových úseků. Podstatná část těchto matic však obsahuje velmi nízké hodnoty, které na celkový výsledek mají nepatrný vliv. Je proto možné je zanedbat a počítat pouze prvky matic v okolí „nejlepší cesty“.

Pro zrychlení výpočtu algoritmem forward-backward byla snaha zaměřena zejména na prořezávání samotné matice hodnot $\alpha$, a to již za jejího výpočtu. Hlavním problémem při stanovení vhodného přístupu bylo nalezení kompromisu mezi vyřezáním co největšího množství nepotřebných hodnot a zachováním všech takových, které ve skutečnosti v dalším výpočtu hrají zásadní roli.

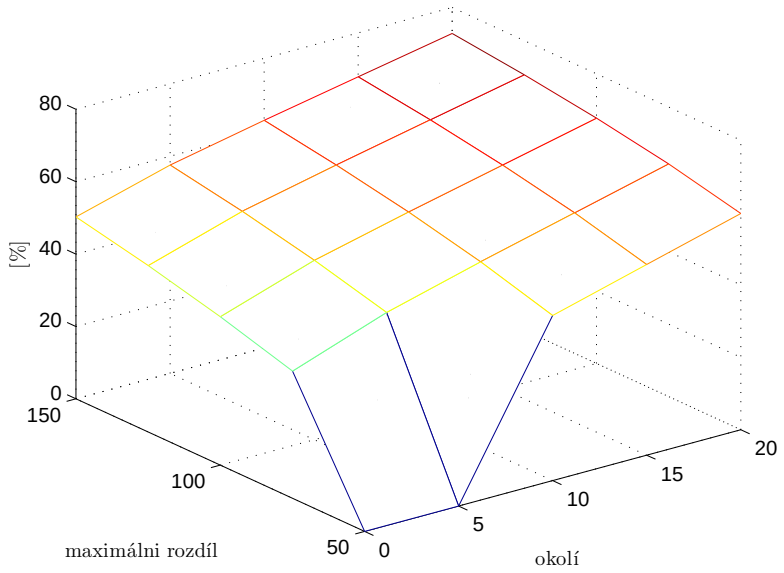
Bylo zvažováno více různých přístupů k prořezávání. Ve všech případech byl vždy nalezen stav $s_i$, kterému v daném časovém okamžiku odpovídala nejvyšší hodnota $\alpha_i(t)$. Nejjednodušší možností poté bylo ponechat pouze ty hodnoty, které se od maxima lišily o méně než stanovenou mez. Zbylé prvky matice byly pak považovány za příliš nízké pravděpodobnosti na to, aby příslušné stavy měly v tomto čase vliv na konečný výsledek. Toto bylo realizováno ponecháním určitého pásu hodnot kolem maxima, přičemž jeho kraje byly určeny jako nejbližší prvky, nesplňující podmínku maximální přípustné velikosti rozdílu. V dalším kroku výpočtu pak byly vždy hodnoty $\alpha$ počítány pouze pro tento pás, rozšířený pouze o následující stav, do kterého je možno přejít. Později byl algoritmus dále pozměněn, aby umožňoval sledování několika takovýchto nejlepších cest.

Komplikací, která se zde vyskytla, byly případy, kdy v blízkosti maxima hodnoty $\alpha$ prudce poklesly, a byly proto vyřezány, načež se ovšem ukázalo, že jimi vede optimální cesta získaná při zpětném běhu. Jelikož prořezaným hodnotám byla automaticky přiřazována nulová pravděpodobnost (reprezentovaná velkou zápornou hodnotou, jelikož bylo pracováno s logaritmy), toto vedlo ke značnému zkreslení výsledku. Tento problém bylo možné řešit dvěma způsoby - dostatečným zvětšením meze stanovující maximální přijatelný rozdíl hodnot oproti maximu, což ovšem vede ke značnému snížení počtu prořezaných hodnot, nebo

přidáním určitého okolí optimální cesty, které je ponecháno bez ohledu na hodnoty $\alpha$. Toto okolí bylo zvoleno jako pevný počet stavů následujících po posledním stavu ponechaném podle kritéria rozdílu oproti maximu.

Aby bylo možno určit vhodnou kombinaci obou zvažovaných přístupů, byl provedeno trénování modelu na všech nahrávkách pro různé kombinace maximálního rozdílu $\alpha$ a velikosti okolí a byly vypočteny průměrné podíly ponechaných hodnot z prvních sedmi iterací trénování. Veškeré výsledky byly porovnávány s během bez prořezávání, aby bylo zaručeno, že nedochází k výrazným odlišnostem.

Obrázek 6.1 zobrazuje graf závislosti průměrného ponechaného podílu matice hodnot $\alpha$ během prvních 7 iterací trénovacího cyklu na hodnotách parametrů, určujících míru prořezávání. Nulové hodnoty představují kombinace parametrů, pro které vlivem prořezávání dospěl algoritmus forward-backward pro některou z nahrávek k zásadně odlišnému výsledku než při jeho použití bez prořezávání. Je očividné, že ačkoliv se zvětšujícím se ponechaným okolím nejlepší cesty roste množství zpracovávaných hodnot, pro obsažení všech hodnot, potřebných k získání správného výsledku tento přístup sám o sobě nepostačuje. Na základě poznatků, získaných z tohoto grafu, bylo nakonec upuštěno od využití okolí nejlepších stavů a prořezávání bylo poté prováděno pouze na základě rozdílů hodnot $\alpha$ oproti maximu.




Obrázek 6.1: Průměrná ponechaná část z matice hodnot $\alpha$ v algoritmu forward-backward

# 7 Optimalizace z hlediska úspěšnosti rozpoznávání

## 7.1 Použitá data a modely

### 7.1.1 Popis nahrávek

Experimenty byly prováděny na množině 62 promluv různých řečníků, mužů i žen, sestávajících pouze z deseti číslovek od nuly do devíti, vyslovovaných v různém pořadí. Kromě těchto číslovek se v nahrávkách nevyskytovala žádná jiná slova.

Jednalo se o telefonní nahrávky, vzorkované frekvencí 8kHz a 16-bitovým kódováním. Pro pozdější experimenty byla rovněž provedena simulace různých druhů zkreslení nahrávek pomocí softwaru Sound Forge Pro 10. Prvním druhem zkreslení byl GSM kodek 6.10, dále simulace charakteristiky stolního mikrofonu. K nahrávkám se simulovaným stolním mikrofonem byl také přimíchán aditivní šum pořízený z jedoucího auta. Šum byl pořízen mikrofonem notebooku, převzorkován z původních 44,1 kHz na 8kHz. Nahrávky byly pořízeny za stálé rychlosti při jízdě po dálnici.

Promluvy byly rozděleny do dvou množin. Prvních 30 nahrávek bylo použito pro trénování akustického modelu a zbylých 32 představovalo testovací množinu.

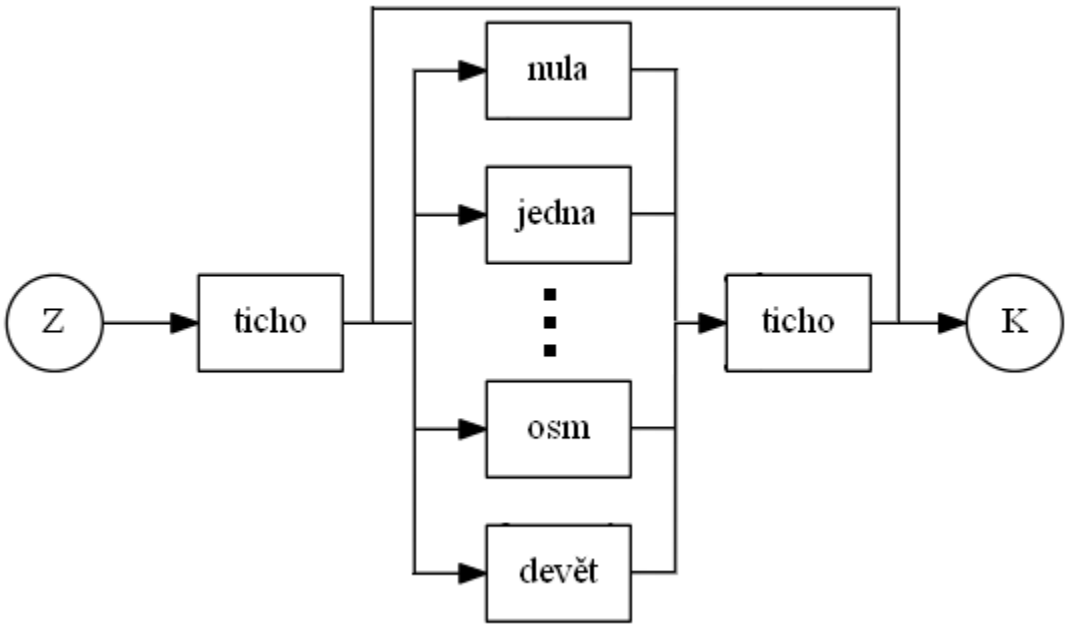
Parametrizace nahrávek byla provedena pomocí metody MFCC. Získávány byly vektory 5 příznaků.

### 7.1.2 Popis použitých modelů

Pro akustické modelování byl použit skrytý Markovův model, obsahující výstupní hustoty pravděpodobnosti představované pouze jednoduchým normálním rozdělením s diagonální kovarianční maticí. Toto zjednodušení umožňuje mimo jiné snadné porovnávání natrénovaných modelů a vypočítaných statistik použitých dat.

Jazykový model je schematicky znázorněn na obrázku 7.1. Všechna slova byla jazykovým modelem ohodnocena stejnou pravděpodobností. Kromě jednotlivých číslovek byly rovněž rozpoznávány neřečové události, rozdělené do dvou kategorií: „dech“ a ostatní rušivé zvuky na pozadí, do výsledné posloupnosti rozpoznaných slov však nebyly zahrnovány.

Jednotlivá slova byla modelována jako celky, přičemž každému slovu připadal stejný počet stavů HMM. Tento počet byl určen experimentálně (viz 7.3). Ticho bylo představováno jediným stavem.




Obrázek 7.1: Schématické znázornění jazykového modelu

## 7.2 Hodnocení úspěšnosti rozpoznávání

Pro vyhodnocení úspěšnosti rozpoznávání slov byl použit výpočet založený na Levenshteinově vzdálenosti.

Levenshteinova vzdálenost ve své základní podobě slouží pro měření rozdílů mezi dvěma řetězci. Představuje minimální počet znaků, které je třeba změnit, vložit nebo odstranit, abychom z prvního řetězce získali druhý.

V tomto případě byl pro každou větu zjišťován počet slov, která byla chybně rozpoznána, přebývala nebo naopak chyběla.

Celková úspěšnost rozpoznávání pak byla vyjádřena podle následujícího vztahu:

$$Acc = \frac{1}{N}\sum_{n=1}^{N}(1 - \frac{L(W_{ref}, W_{rec})}{N_{W_{ref}}}) \cdot 100\%, \tag{7.1}$$

kde $N$ je počet vět testovací množiny, $W_{ref}$ je posloupnost skutečných slov věty, $W_{rec}$ je posloupnost slov, která byla rozpoznána, $N_{W_{ref}}$ je počet slov v posloupnosti $W_{ref}$ a $L(W_{ref}, W_{rec})$ představuje Levenshteinovu vzdálenost posloupností $W_{ref}$ a $W_{rec}$.

Pro podrobnější zhodnocení kvality rozpoznávání lze rovněž využít program HResults, který je součástí balíku nástrojů HTK (Hidden Markov Model Toolkit). Tento program udává následující hodnoty [12]:

- $N$ - celkový počet slov referenční posloupnosti,
- $S$ - počet záměn správného slova za jiné,
- $D$ - počet vynechaných slov,
- $I$ - počet slov vložených navíc,
- $Corr$ - podíl správně rozpoznaných slov, bez ohledu na slova navíc,

$$Corr = \frac{N - D - S}{N} \cdot 100\% \tag{7.2}$$

- $Acc$ - celková úspěšnost rozpoznávání

$$Acc = \frac{N - D - S - I}{N} \cdot 100\% \tag{7.3}$$

Úspěšnost rozpoznávání počítaná pomocí Levenshteinova algoritmu odpovídá hodnotě $Acc$ poskytované HResults.

## 7.3 Určení počtu stavů připadajících na každé slovo

Prvním parametrem, který bylo během experimentů třeba určit, byl optimální počet stavů modelu, připadajících na každé slovo. Vyšší počet stavů může vést ke zlepšení přesnosti rozpoznávání, zejména v případě velké podobnosti některých slov, avšak zároveň zvyšuje potřebnou dobu výpočtu.

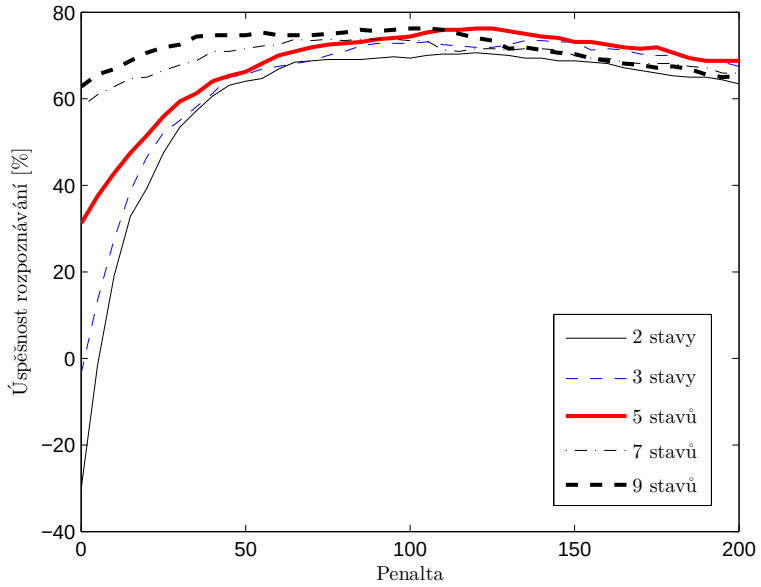
Vhodný počet stavů každého slova byl hledán na základě úspěšnosti rozpoznávání slov.

| počet stavů | úspěšnost rozpoznávání |
|---|---|
| 2 | -30,0% |
| 3 | -3,44% |
| 4 | 15,9% |
| 5 | 31,2% |
| 6 | 38,4% |
| 7 | 58,1% |
| 8 | 56,6% |
| 9 | 62,8% |

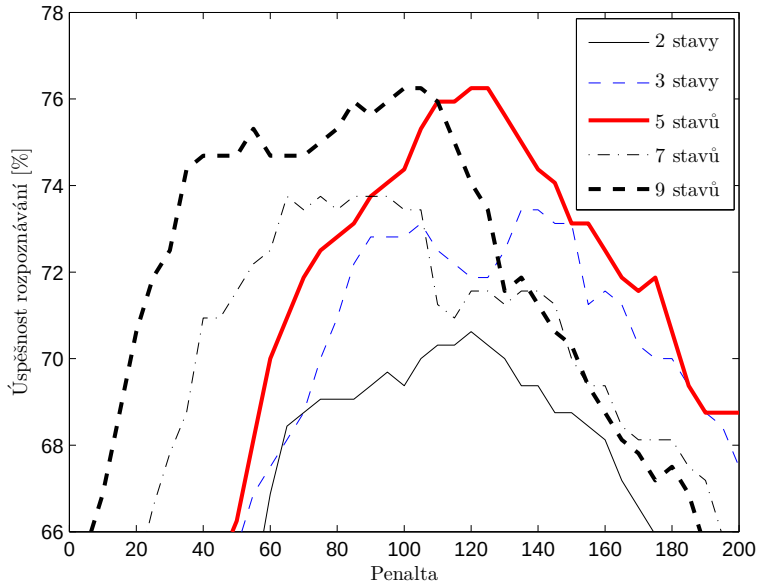
Tabulka 7.1: Úspěšnost rozpoznávání pro různé počty stavů připadajících na jedno slovo

Z tabulky 7.1, uvádějící získanou úspěšnost rozpoznávání pro různé počty stavů každého slova, se zdá, že čím vyšší počet stavů zvolíme, tím vyšší je úspěšnost rozpoznávání.

Po zavedení penalizace slov (viz část 7.5) se však ukázalo, že předchozí domněnka není zcela pravdivá. Obrázky 7.2 a 7.3 zobrazují graf závislosti úspěšnosti rozpoznání na velikosti penalty vložení pro různé počty stavů připadajících na každé slovo. Pro větší přehlednost jsou zobrazeny křivky pouze pro některé z hodnot. Graf ukazuje, že ačkoliv pro nízké hodnoty penalty vložení úspěšnost rozpoznávání skutečně roste s počtem stavů každého slova, pro dostatečně vysokou penaltu jsou jednotlivé úspěšnosti mnohem vyrovnanější a uvedená závislost zcela neplatí. Nejlepší úspěšnosti rozpoznávání bylo dosaženo při použití pěti nebo devíti stavů modelu pro každé slovo. Jelikož vyšší počet stavů znamená delší dobu výpočtu během trénování i dekódování, jako optimální byla zvolena nižší z těchto dvou hodnot, tedy 5 stavů na slovo.




Obrázek 7.2: Celková úspěšnost rozpoznávání slov pro různé velikosti penalty vložení a různé počty stavů HMM připadající na jedno slovo




Obrázek 7.3: Celková úspěšnost rozpoznávání slov pro různé velikosti penalty vložení a různé počty stavů HMM připadající na jedno slovo - detail

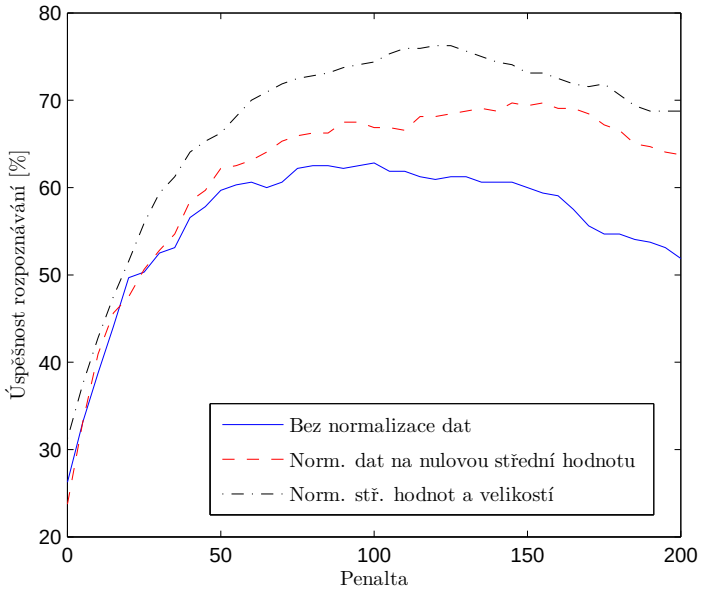
## 7.4 Normalizace příznakových vektorů

Dalším důležitým rozhodnutím bylo, zda během experimentů provádět v rámci parametrizace normalizaci příznakových vektorů.

Nabízely se tři možnosti:

1. Žádná normalizace
2. Normalizace středních hodnot (Normalizovaná data mají nulovou celkovou střední hodnotu)
3. Normalizace středních hodnot a velikostí (Normalizovaná data mají nulovou celkovou střední hodnotu a jednotkovou maximální velikost)

Jelikož předchozí experimenty byly prováděny pro data, normalizovaná dle třetí možnosti, a zároveň se ukázalo, že tato možnost vede k nejvyšší úspěšnosti rozpoznávání ze všech tří zvažovaných (znázorněno na obr. 7.4, penalta je zavedena v části 7.5), byla tato varianta zvolena i pro další experimenty.




Obrázek 7.4: Úspěšnost rozpoznávání pro různá nastavení normalizace dat

## 7.5 Penalizace slov

Srovnáním rozpoznaných posloupností slov s referenčním přepisem promluv bylo zjištěno, že je rozpoznáváno výrazně větší množství slov, než by správně mělo být. Pro omezení počtu slov rozpoznaných „navíc“ je proto vhodné zavést penalizaci slov. Ta se během dekódování aplikuje tím způsobem, že vždy, když model přejde do stavu odpovídajícího novému slovu, sníží se výsledná logaritmická pravděpodobnost o určitou hodnotu. Toto by mělo akustický model vést k tomu, aby namísto rychlého procházení velkého množství velmi krátkých „slov“ volil cestu, při které v jednotlivých slovech nebo v pauze stráví delší dobu.

Výši penalizace je třeba vhodně stanovit. Měla by být dostačovat k tomu, abychom odstranili co nejvíce přebytečných nesprávných slov. Zároveň však hrozí, že pokud zvolíme příliš vysokou hodnotu, dojde ke ztrátě nadměrného množství slov správných.

Máme-li testovací množinu nahrávek, dostatečně reprezentující promluvy, které bude třeba rozpoznávat, a referenční přepisy těchto nahrávek, lze velikost penalty stanovit experimentálně na základě různých kritérií.

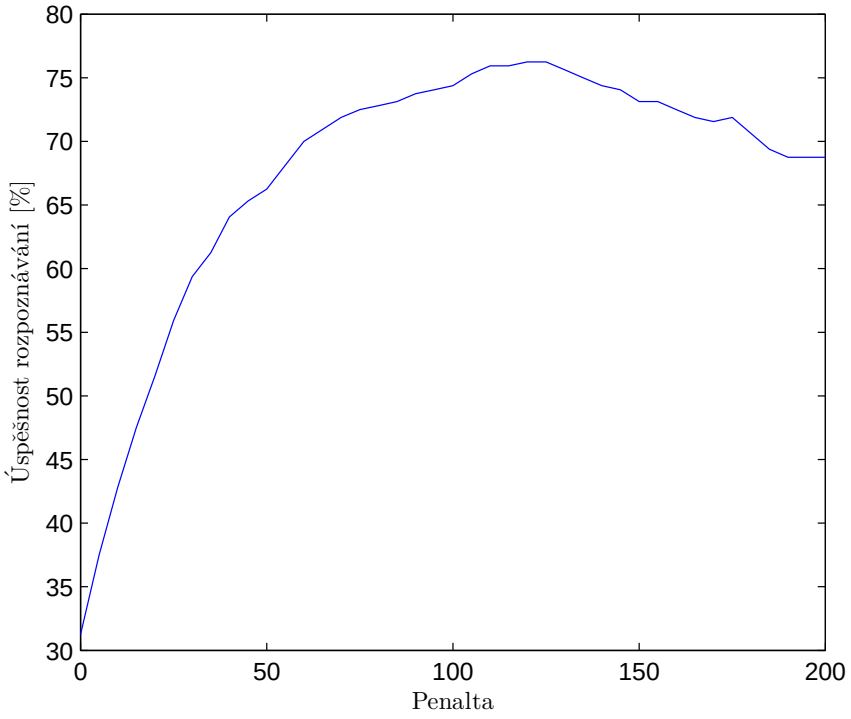
V reálných systémech rozpoznávání je třeba kromě velikosti penalty vložení slov nalézt rovněž vhodnou váhu jazykového modelu. Je tak vyžadováno hledání optimálního nastavení ve dvourozměrném prostoru, což značně zvyšuje časovou náročnost tohoto úkolu. V tomto případě však vzhledem k jednoduchosti úlohy nebylo určování váhy jazykového modelu zapotřebí.

### 7.5.1 Stanovení penalty na základě úspěšnosti rozpoznávání

Jelikož úspěšnost rozpoznání představuje vedle rychlosti výpočtu hlavní parametr, na kterém při rozpoznávání řeči záleží, je logické, že bychom tento parametr mohli použít pro stanovení optimální hodnoty penalizace přechodu modelu do stavu nového slova.

Byla proto zjišťována závislost úspěšnosti rozpoznávání na zvolené hodnotě penalty. Na obrázku 7.5 je znázorněn výsledný graf. Ten napovídá, že s rostoucí penaltou úspěšnost nejprve stoupá v důsledku snižujícího se počtu přebytečných rozpoznaných slov, avšak v určitém okamžiku ztráta způsobená chybějícími správnými slovy převýší tento zisk, a úspěšnost rozpoznávání poté začne klesat. Optimální hodnota penalty pro použitá data

se pak nachází přibližně v intervalu 110-125, na kterém je úspěšnost nejvyšší.




Obrázek 7.5: Závislost celkové úspěšnosti rozpoznávání slov na zvolené výši penalizace přechodu modelu do stavů nového slova

Tento způsob stanovení výše penalizace má ovšem zásadní nevýhodu. Jelikož je třeba stanovit maximum funkce, která navíc může mít řadu lokálních extrémů, správné určení optimální hodnoty vyžaduje provedení rozpoznávání pro relativně velké množství různých penalt. Tento výpočet poté zabere poměrně dlouhou dobu.

Vhodnější by tedy bylo zvolit jako rozhodující takový parametr, jehož závislost na velikosti penalty je monotónní, ideálně lineární funkcí.

### 7.5.2 Stanovení penalty na základě statistik z HResults

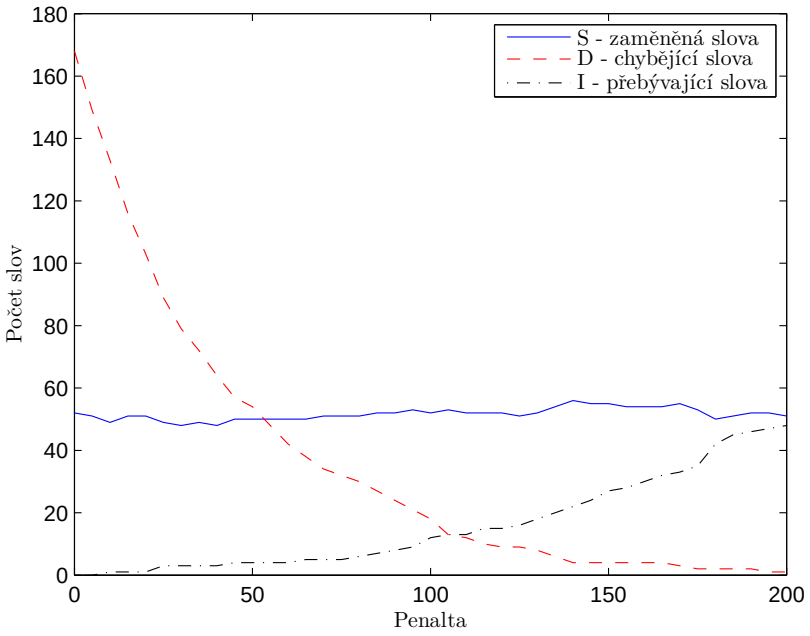
Jako další možnost určení vhodné velikosti penalty bylo zvažováno využití statistik poskytnutých programem HResults, podrobněji popsaných v části 7.2.

Graf 7.6 ukazuje počty zaměněných ($S$), vložených ($I$) a vynechaných ($D$) slov v rozpo-

znané posloupnosti oproti referenčnímu přepisu promluv v závislosti na použité hodnotě penalty. Celkový počet slov přitom byl 320 a neřečové události nejsou zahrnuty.

Dle očekávání s rostoucí penaltou klesá počet vložených slov, ovšem rovněž stoupá počet slov vynechaných. Počet zaměněných slov se ve srovnání výrazně nemění, optimální hodnota penalty se proto pravděpodobně nachází v oblasti kolem průsečíku ostatních dvou křivek, který v tomto případě odpovídá přibližně hodnotě 105. Toto poměrně dobře souhlasí se zjištěním z bodu 7.5.1, kde bylo z grafu vyčteno, že úspěšnost rozpoznávání je nejlepší pro hodnoty penalizace z intervalu 110-125.

Ačkoliv tedy tímto kritériem nemusíme nalézt zcela přesně optimální hodnotu penalizace z hlediska úspěšnosti rozpoznávání na testovací množině nahrávek, zdá se, že vede k jejímu poměrně dobrému odhadu. Není přitom nutné provádět rozpoznávání pro tak velký počet různých hodnot, jelikož obě křivky jsou monotónní, a jejich průsečík lze tak s využitím interpolace na základě několika málo bodů.

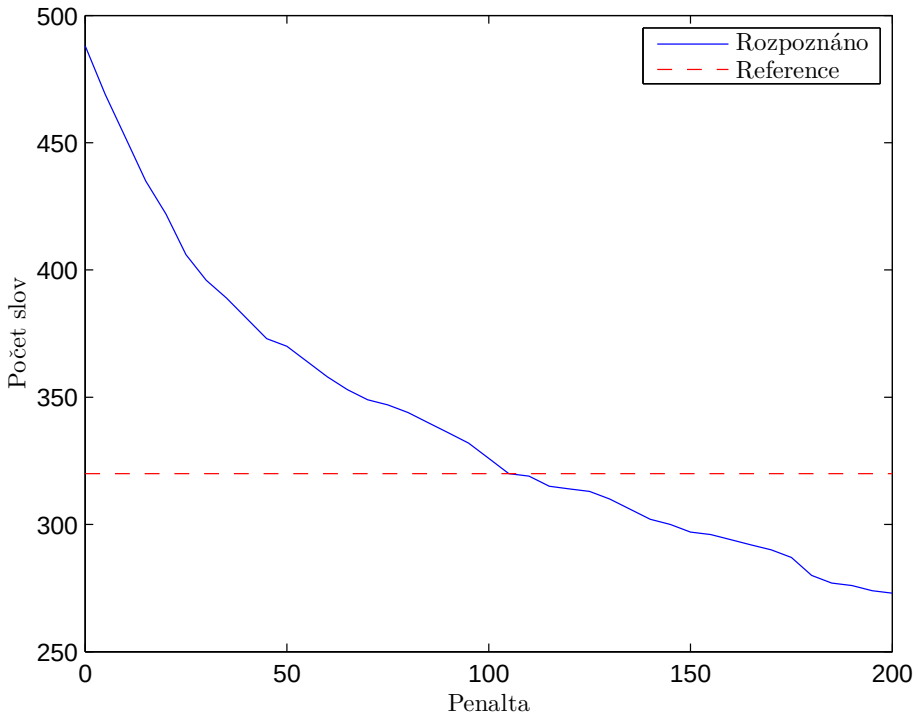



Obrázek 7.6: Počty zaměněných, chybějících a přebývajících slov v rozpoznané posloupnosti slov oproti referenčnímu přepisu, v závislosti na zvolené penaltě

### 7.5.3 Stanovení penalty na základě počtu rozpoznaných slov

Jelikož průsečík hodnot $D$ a $I$, zvolený v bodu 7.5.2 jako vhodná velikost penalty vložení, představuje situaci, kdy se počet slov vložených navíc rovná počtu slov chybně odstraněných, takto formulované kritérium je ve své podstatě pouhým nalezením hodnoty, pro kterou je celkový počet rozpoznaných slov roven skutečnému. Proto je možné omezit se pouze na tento parametr a nezabývat se zjišťováním hodnot $S$, $I$ a $D$.

Příklad odpovídajícího grafu je znázorněn na obrázku 7.7. Je zde ukázána závislost celkového počtu rozpoznaných slov na použité penaltě vložení a rovněž skutečný celkový počet slov v nahrávkách. Obě křivky se skutečně navzájem protínají v bodu odpovídajícím stejné penaltě vložení, jaká byla získána v části 7.5.2.




Obrázek 7.7: Počty rozpoznaných slov pro různé hodnoty penalty a jejich skutečný počet

# 8 Analýza výsledků rozpoznávání

## 8.1 Vyhodnocení z hlediska konkrétních stavů HMM

### 8.1.1 Srovnání s referenční posloupností stavů

Aby mohlo být rozpoznávání lépe vyhodnoceno na úrovni jednotlivých stavů, bylo pro srovnání provedeno rozpoznávání rovněž se znalostí referenčního přepisu. Úkolem pak bylo pro konkrétní posloupnost slov získat nejpravděpodobnější odpovídající posloupnost stavů. Ta byla dále považována za správnou.

Tato referenční posloupnost stavů pak umožňuje zejména identifikovat nejčastější stavy, u kterých dochází k chybnému rozpoznání.

Příklad srovnání referenční a rozpoznané posloupnosti stavů je zobrazen na obrázku 8.1. Grafy znázorňují posloupnosti stavů odpovídajících jedné z promluv, a to jak pro výpočet s penalizací slov tak bez ní. První graf, odpovídající rozpoznávání bez penalizace, vykazuje značně větší rozdíly mezi oběma posloupnostmi stavů, než graf druhý. Tři ze slov zde byla rozpoznána chybně, další čtyři byla rozdělena do dvou kratších slov následujících po sobě a objevují se zde dvě slova v oblastech referenčním výpočtem rozpoznaných jako ticho. V druhém grafu se naopak žádná slova navíc neobjevují, ovšem lze zde spatřit, že poslední slovo bylo vlivem penalizace ztraceno. Počet záměn pak je stejný jako v prvním případě.

Posloupnost stavů z druhého grafu odpovídá posloupnosti slov

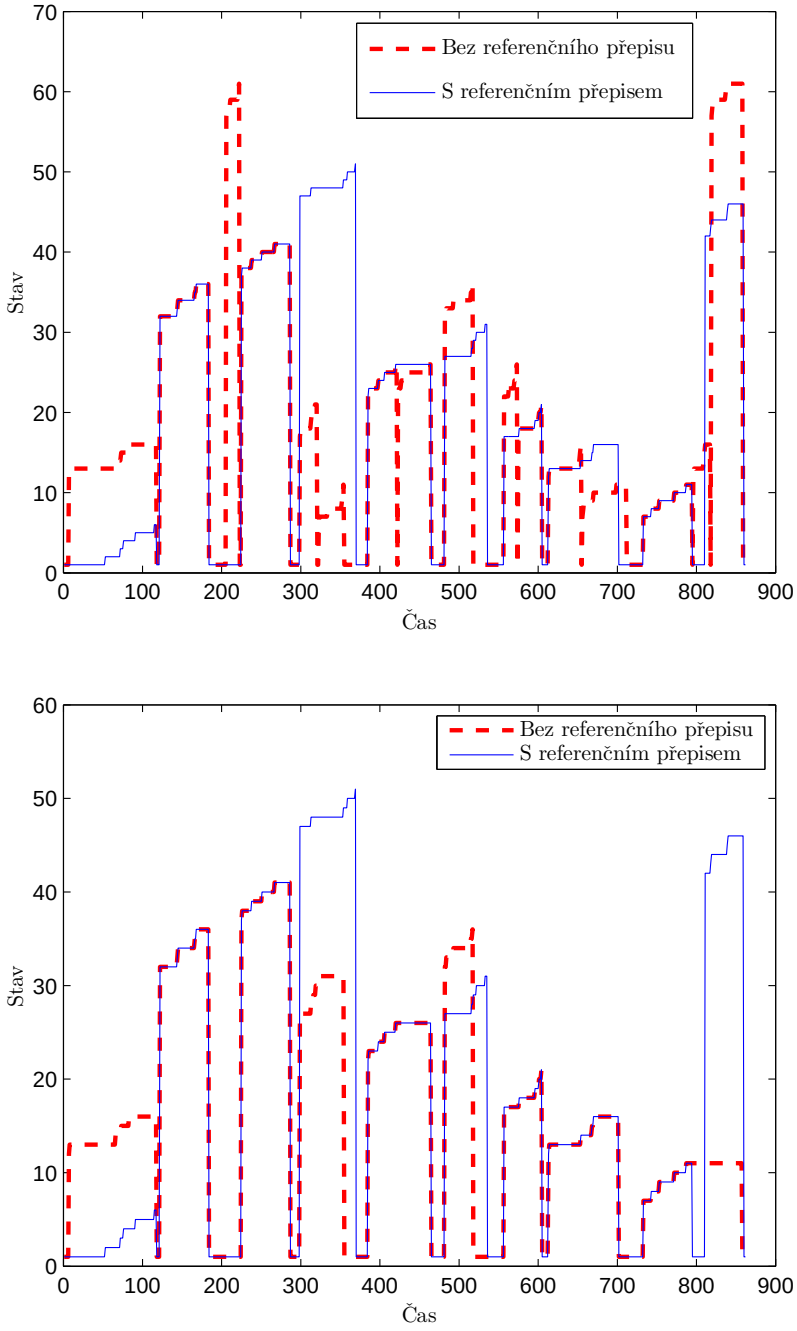
„dva, šest, sedm, pět, čtyři, šest, tři, dva, jedna“.

Ve skutečnosti však v nahrávce byla pronesena promluva

„*nula*, šest, sedm, *devět*, čtyři, *pět*, tři, dva, jedna, *osm*“.

Došlo tu tedy k záměně slov „nula“ za „dva“, „devět“ za „pět“, „pět“ za „šest“, a ke ztrátě slova „osm“.

Přehled stavů odpovídajících jednotlivým slovům lze nalézt v tabulce 8.1 na straně 52.

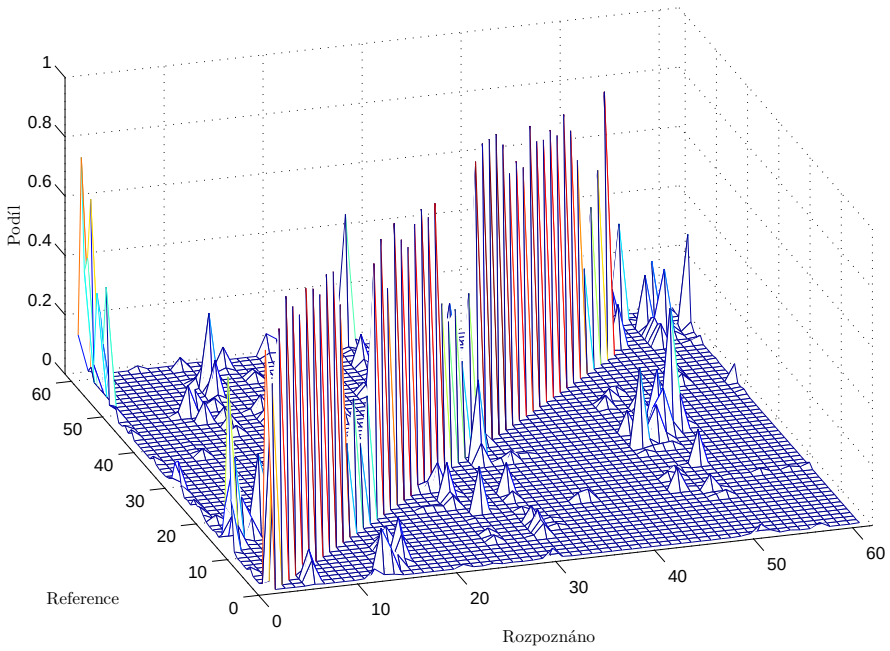



Obrázek 8.1: Srovnání nejpravděpodobnější posloupnosti stavů pro jednu z vět testovací množiny, určené dekódováním s referenčním přepisem a bez něj, bez penalizace slov (horní graf) a s penalizací (spodní graf)

### 8.1.2 Určení nejčastěji zaměňovaných stavů

Pro snadné určení nejčastějších záměn byla na základě rozpoznaných a referenčních posloupností stavů vytvořena matice $P$, znázorněná na obrázku 8.2.

Každý její prvek $p(i, j)$ představuje relativní podíl časových okamžiků rozpoznaných jako stav $s_j$ ze všech takových, které referenční posloupností byly identifikovány jako stav $s_i$. Diagonála tedy představuje správně rozpoznané stavy, větší hodnoty mimo ni pak odpovídají těm dvojicím stavů, mezi kterými často dochází k záměně. V ideálním případě, kdy by rozpoznaná posloupnost stavů naprosto souhlasila s referenční, by pak tato matice byla jednotková.



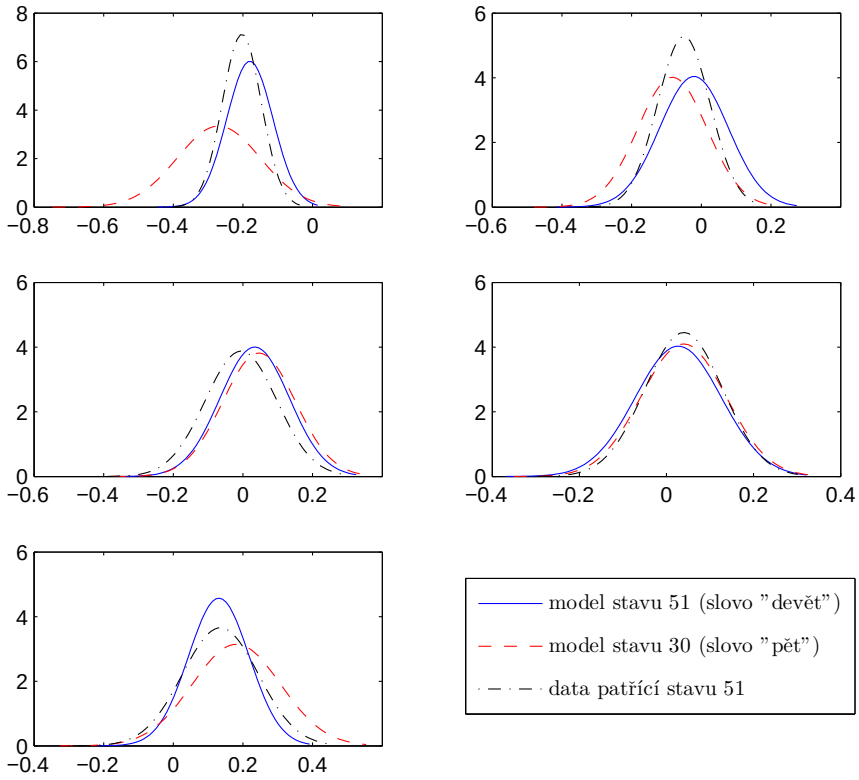

Obrázek 8.2: Relativní podíly časových okamžiků, odpovídajících konkrétnímu stavu modelu, které byly rozpoznány jako stav jiný

V grafu lze spatřit několik problematických oblastí, představujících velmi časté záměny. Pro rozmezí stavů odpovídající slovům „dva“, „pět“ a „devět“ a neřečovým událostem se na diagonále vyskytují hodnoty často menší než jedna polovina. To znamená, že ve více než polovině případů jsou tato slova chybně rozpoznána. Vyhledáním vysokých hodnot v oblasti mimo diagonálu pak zjistíme, že slovo „dva“ bývá zaměňováno za slovo „nula“

a slovo „pět“ za „devět“ a naopak. Obojí je způsobeno značnou podobností těchto dvojic slov.

Neřečové události bývají velmi často označovány jako ticho, avšak toto je pravděpodobně pouze následek vysoké penalizace a není příliš podstatné, jelikož neřečové události ve výsledné posloupnosti slov, která je vyhodnocována, nejsou zahrnuty.

Obrázek 8.3 zobrazuje srovnání křivek znázorňujících natrénované výstupní hustoty pravděpodobností HMM pro dva z často navzájem zaměňovaných stavů, náležející slovům „devět“ a „pět“, a rozložení dat z testovací množiny, které podle referenční posloupnosti odpovídají prvnímu z nich. Každý z pěti grafů představuje jeden příznak. Obrázek ukazuje značnou podobnost obou stavů modelu, která způsobuje chybné rozpoznání některých dat.




Obrázek 8.3: Srovnání výstupních hustot pravděpodobností dvou stavů HMM a dat, náležejících jednomu z nich, pro všech 5 dimenzí příznakových vektorů, jedná se o dva nejčastěji navzájem zaměňované stavy ze slov „devět“ a „pět“

| stavy | slovo |
|---|---|
| 1 | (ticho) |
| 2-6 | „nula“ |
| 7-11 | „jedna“ |
| 12-16 | „dva“ |
| 17-21 | „tři“ |
| 22-26 | „čtyři“ |
| 27-31 | „pět“ |
| 32-36 | „šest“ |
| 37-41 | „sedm“ |
| 42-46 | „osm“ |
| 47-51 | „devět“ |
| 52-56 | (zvuk na pozadí) |
| 57-61 | (dech) |

Tabulka 8.1: Přehled indexů stavů náležejících jednotlivým slovům

## 8.2 Srovnání středních hodnot a variancí modelu a dat

Aby akustický model dobře sloužil k rozpoznávání zadaných promluv, je třeba, aby byla použita vhodná trénovací data. V optimálním případě by měly střední hodnoty a rozptyly natrénovaného modelu co nejlépe odpovídat středním hodnotám a rozptylům příznakových vektorů rozpoznávaných promluv. Pokud toto není splněno, existuje několik situací, které mohou nastat.

Jednou z možností je situace, kdy sobě odpovídají střední hodnoty, avšak výrazně se liší rozptyly. Toto může nastat, jestliže jedna množina nahrávek obsahuje promluvy pouze jednoho nebo několika málo podobných řečníků, zatímco v druhé je zastoupeno větší množství odlišných. Jestliže rozptyl modelu je menší než rozptyl rozpoznávaných dat, znamená to, že model byl natrénován příliš konkrétně a nepostihuje všechna rozpoznávaná data. V opačném případě byl naopak natrénován příliš obecně. U této druhé varianty je možné model pomocí adaptace lépe přizpůsobit konkrétním datům.

Další možnost představuje případ, kdy jsou podobné rozptyly, ale naopak odlišné střední hodnoty. K tomuto může dojít při přítomnosti konstantního šumu na pozadí rozpoznávaných promluv nebo pokud byl model natrénován příliš konkrétně a hlas řečníka z rozpoznávané promluvy se výrazně liší. Příkladem může být nahrávka telefonátu z jedoucího automobilu, nebo použití modelu, natrénovaného na mužský hlas, pro rozpoznání řeči dítěte.

Nejkomplikovanějším případem pak je kombinace obou těchto variant.

Aby bylo možno určit, o který případ se pro danou dvojici trénovacích a testovacích dat jedná, byly vždy počítány rozdíly středních hodnot a variancí natrénovaného akustického modelu a dat pro odpovídající si stavy.

Hodnoty $dm$, vyjadřující rozdíl středních hodnot, a $dv$, vyjadřující rozdíl variancí, byly počítány podle vztahů

$$dm = \sqrt{E_i \left\{ E_j \left[ \frac{(\mu_{ij} - m_{ij})^2}{c_{ij}} \right] \right\}} \tag{8.1}$$

$$dv = \sqrt{E_i \left\{ E_j \left[ \frac{v_{ij}}{c_{ij}} \right] \right\}}, \tag{8.2}$$

kde $\mu_{ij}$ značí $j$-tou složku střední hodnoty výstupní hustoty pravděpodobnosti stavu $s_i$ HMM, $c_{ij}$ odpovídající prvek kovarianční matice, $m_{ij}$ značí $j$-tou složku střední hodnoty dat, příslušejících ke stavu $s_i$ a $v_{ij}$ je variance těchto dat.

Pro data, jejichž střední hodnoty a variance přesně odpovídají natrénovanému modelu by $dm$ mělo být rovno nule a $dv$ rovno jedné. Jestliže je $dv$ větší než jedna, znamená to, že variance dat je vyšší než variance modelu, model je tedy natrénován příliš konkrétně. $dv$ menší než jedna naopak značí příliš obecně natrénovaný model.

Pro použité trénovací a testovací množiny nahrávek byly získány tyto hodnoty parametrů $dm$ a $dv$:

$$dm = 0,316$$
$$dv = 1,017$$

## 8.3 Vliv zkreslení dat na kvalitu rozpoznávání

Jelikož trénovací i testovací množina pocházely ze stejné sady poměrně kvalitních nahrávek, rozdíly mezi těmito množinami byly malé. V praktickém využití rozpoznávání řeči se však často rozpoznávané nahrávky mohou od trénovací množiny značně lišit. Proto byl zkoumán vliv zkreslení nahrávek na kvalitu rozpoznávání, nejprve pomocí šumu přidaného přímo k příznakovým vektorům, a to pro dva speciální případy. Prvním z nich bylo přičtení šumu s nulovou střední hodnotou a nenulovou variancí, druhým naopak změna příznaků o konstantní hodnotu. V pozdějších experimentech byly rovněž zkresleny samotné nahrávky.

### 8.3.1 Přidání šumu s různou variancí k příznakovým vektorům

Použitá data byla zkreslena přičtením náhodného šumu k příznakovým vektorům nahrávek. Jednalo se o šum s normálním pravděpodobnostním rozložením, nulovou střední hodnotou a variancí rovnou $K$-násobku celkové variance původních dat, kde $K$ představovalo hodnoty z intervalu od 0 do 1. Toto tedy v podstatě představovalo zvětšení variance daných dat. Následně bylo každou sadu dat provedeno natrénování akustického modelu a rozpoznávání promluv z testovací množiny pro následující tři možnosti: zašumění trénovacích dat, zašumění testovacích dat a zašumění trénovacích i testovacích dat.

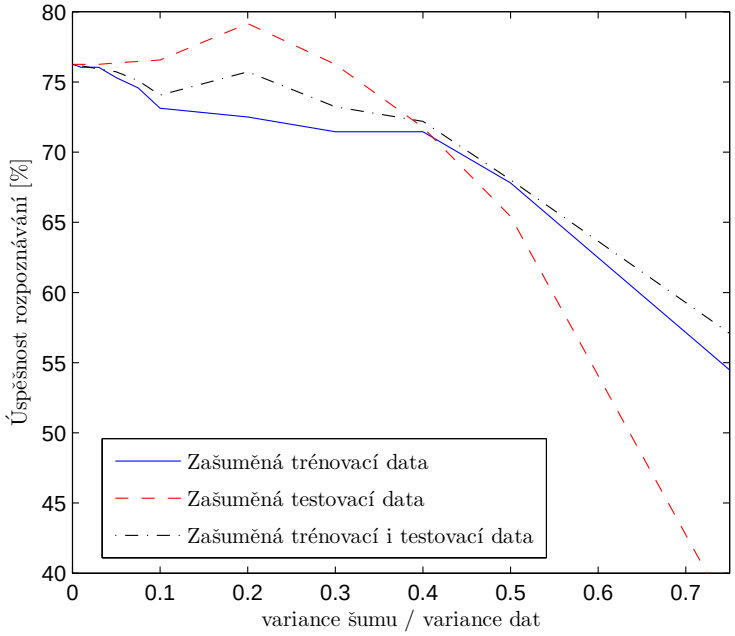
Při zpracování výsledků se ukázalo, že ačkoliv pro velmi zašuměná data kvalita rozpoznávání značně klesá, za určitých okolností může naopak dojít až k mírnému zlepšení úspěšnosti rozpoznávání. Nejvýrazněji tento jev nastal v případě zašumění pouze testovacích dat. Tento fakt napovídá tomu, že variance modelu pro původní trénovací data byly pravděpodobně přinejmenším pro některá slova větší než variance původních testovacích dat, což bylo přidáním šumu napraveno, a tato nová data tak lépe odpovídají modelu.

Výsledný graf znázorňující závislost úspěšnosti rozpoznávání na varianci přidaného šumu je zobrazen na obrázku 8.4. Jednotlivé hodnoty byly získány zprůměrováním výsledků pro tři různé realizace šumu.

Následně byly rovněž porovnány střední hodnoty a variance modelů a dat pomocí parametrů $dm$ a $dv$ definovaných v části 8.2. Hodnoty $dv$, vyjadřující srovnání variancí, jsou obsaženy v tabulce 8.2. Ta ukazuje, že bez aplikace šumu jsou variance dat i modelu po zprůměrování přibližně stejné, avšak s rostoucí variancí šumu se tento poměr skutečně odpovídajícím způsobem mění. Pro zašuměná testovací data je jejich variance oproti varianci

natrénovaného modelu výrazně větší, při zašumění dat trénovacích se naopak hodnota *dv* snižuje. V případě, že zašumíme obě sady dat šumem s podobnou variancí, hodnota *dv* zůstává přibližně stejná.

Jelikož použitý šum měl nulovou střední hodnotu, střední hodnoty se nijak významně neměnily. Tabulka pro parametr *dm* zde proto není uvedena.




Obrázek 8.4: Závislost úspěšnosti rozpoznávání na varianci přidaného šumu pro zašuměná trénovací, testovací a trénovací i testovací data, průměry ze tří realizací

### 8.3.2 Přidání šumu s různou střední hodnotou k příznakovým vektorům

Pro srovnání byl předchozí experiment proveden rovněž pro přidaný šum s nulovou variancí a nenulovou střední hodnotou. Jednalo se tedy o pouhé přičtení konstanty k příznakovým vektorům, což může znázorňovat konstantní neměnný zvuk na pozadí nahrávek. Velikost této konstanty byla opět stanovena relativně k varianci dat.

Parametrizovaná data jsou pětidimenzionální, a proto je vysoce pravděpodobné, že rozdíly středních hodnot mezi promluvami stejných slov v trénovací a testovací množině se pro

| | zašuměná data | | |
|---|---|---|---|
| K | testovací | trénovací | všechna |
| 0 | 1.017 | 1.017 | 1.017 |
| 0.01 | 1.017 | 1.017 | 1.017 |
| 0.03 | 1.018 | 1.016 | 1.017 |
| 0.05 | 1.022 | 1.014 | 1.019 |
| 0.075 | 1.027 | 1.012 | 1.022 |
| 0.1 | 1.035 | 1.008 | 1.025 |
| 0.2 | 1.088 | 0.981 | 1.037 |
| 0.3 | 1.177 | 0.921 | 1.032 |
| 0.4 | 1.283 | 0.880 | 1.043 |
| 0.5 | 1.415 | 0.825 | 1.047 |
| 0.75 | 1.800 | 0.688 | 1.036 |
| 1 | 2.213 | 0.572 | 1.020 |

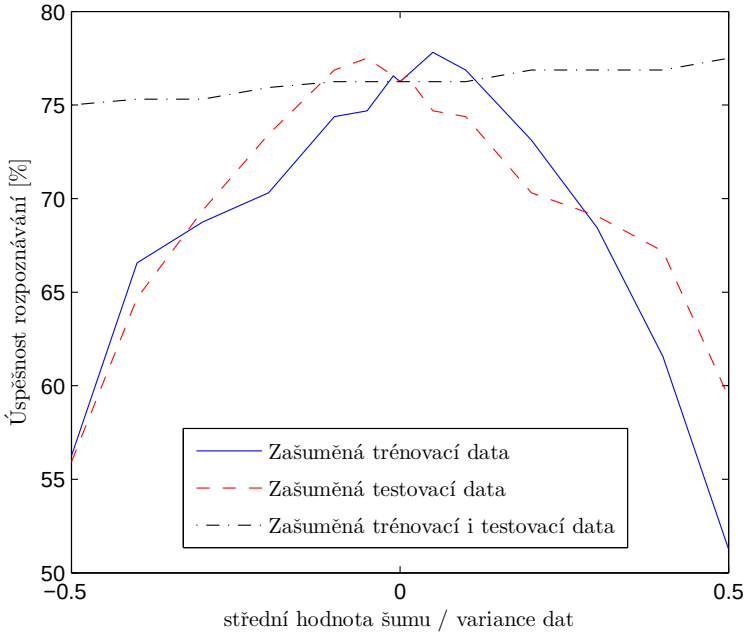
Tabulka 8.2: Srovnání variancí modelu a dat pomocí parametru *dv* pro přidaný šum s nulovou střední hodnotou a variancí rovnou $K$-násobku variance původních dat

jednotlivé dimenze značně liší. Proto jakékoliv zlepšení podobnosti dat a natrénovaného modelu, ke kterému v důsledku přičtené konstanty dojde v jedné dimenzi je s velkou pravděpodobností vyváženo zhoršením v dimenzi jiné. Lze tedy očekávat, že na rozdíl od předchozího experimentu zde v důsledku přidaného šumu ke znatelnému zlepšení úspěšnosti rozpoznávání nedojde.

Jelikož posloupnosti příznakových vektorů jednotlivých nahrávek jsou normalizované na nulovou střední hodnotu a jednotkovou maximální velikost, celková variance dat každé z množin je velmi podobná, a tedy se příliš neliší ani střední hodnoty šumu aplikovaného na trénovací a testovací množinu dat. Z toho poté plyne, že pro případ zašuměných obou množin by výsledná úspěšnost rozpoznávání neměla příliš záviset na podílu střední hodnoty šumu a variance dat. Vedle tohoto by rovněž mělo platit, že přičtení určité hodnoty k datům trénovací množiny má stejný efekt jako její odečtení od množiny testovací.

Graf znázorněný na obrázku 8.5 dokládá, že získané výsledky odpovídaly očekáváním.

Opět byly také porovnány střední hodnoty a variance modelů a dat pomocí parametrů *dm* a *dv*. Vzhledem ke konstantní velikosti šumu tentokrát zůstaly nezměněny variance, došlo však k rozdílům ve středních hodnotách. Tabulka 8.3 proto obsahuje přehled hodnot *dm* pro jednotlivé kombinace trénovacích a testovacích dat. Podobně jako tomu u šumu s různou variancí bylo u hodnot *dv*, i zde se hodnoty získané pro případ zašuměných trénovacích i testovacích dat téměř nemění.




Obrázek 8.5: Závislost úspěšnosti rozpoznávání na střední hodnotě přidaného šumu pro zašuměná trénovací, testovací a trénovací i testovací data

| | zašuměná data | | |
|---|---|---|---|
| $K$ | testovací | trénovací | všechna |
| -0.5 | 1.044 | 1.055 | 0.318 |
| -0.4 | 0.857 | 0.864 | 0.317 |
| -0.3 | 0.677 | 0.680 | 0.317 |
| -0.2 | 0.511 | 0.509 | 0.316 |
| -0.1 | 0.376 | 0.372 | 0.316 |
| -0.05 | 0.333 | 0.330 | 0.316 |
| -0.01 | 0.317 | 0.316 | 0.316 |
| 0 | 0.316 | 0.316 | 0.316 |
| 0.01 | 0.316 | 0.317 | 0.316 |
| 0.05 | 0.329 | 0.333 | 0.316 |
| 0.1 | 0.370 | 0.378 | 0.316 |
| 0.2 | 0.502 | 0.518 | 0.316 |
| 0.3 | 0.667 | 0.689 | 0.317 |
| 0.4 | 0.847 | 0.874 | 0.318 |
| 0.5 | 1.033 | 1.066 | 0.318 |

Tabulka 8.3: Srovnání středních hodnot modelu a dat pomocí parametru $dm$ pro zkreslení dat realizované posunem příznakových vektorů o $K$-násobek variance původních dat

### 8.3.3 Zkreslení samotných nahrávek

Vliv zašumění dat na úspěšnost rozpoznávání byl zkoumán již v částech 8.3.1 a 8.3.2. Nyní však již šum nebude dodatečně aplikován až na příznakové vektory získané parametrizací promluv, jako tomu bylo předtím, ale již samotné nahrávky promluv jsou oproti předchozím zkreslené.

K tomuto účelu byly použity čtyři sady nahrávek stejných promluv:

- GSM - Původní nahrávky, u kterých byla provedena ztrátová komprese pomocí formátu GSM 06.10
- mic - Nahrávky zkreslené simulací charakteristiky stolního mikrofonu
- SNR0 - Nahrávky se simulovaným stolním mikrofonem s přidaným šumem z auta s odstupem 0 dB
- SNR10 - Nahrávky se simulovaným stolním mikrofonem s přidaným šumem z auta s odstupem 10 dB

Trénování bylo prováděno pro jednotlivé zašuměné sady zvlášť, pro všechny jejich nahrávky dohromady, a pro původní nezkreslené nahrávky. Vždy se jednalo o všech 62 promluv, nikoliv jen o původních 30 trénovacích, jako tomu bylo u většiny ostatních experimentů.

Stejných šest sad dat bylo rovněž následně rozpoznáváno, a to pro všechny možné kombinace trénovacích a testovacích dat.

Pro každou z kombinací byly spočítány úspěšnosti rozpoznávání a porovnány střední hodnoty a variance modelů a dat pomocí vzorců uvedených v části 8.2. Výsledky jsou obsaženy v tabulkách 8.4, 8.5 a 8.6.

První tabulka například naznačuje, že přidaný šum z automobilu, zejména varianta SNR0, vede ze zkoumaných možností k nejhorší kvalitě rozpoznávání. Ze třetí tabulky zjistíme proč - je zde výrazný rozdíl ve středních hodnotách oproti ostatním množinám nahrávek. Rozdíly variancí nejsou zdaleka tak výrazné, toto však pravděpodobně lze z velké míry přičíst použité normalizaci dat.

| Úspěšnost rozpoznávání [%] | | | | | | |
|---|---|---|---|---|---|---|
| | rozpoznávané nahrávky | | | | | |
| trénovací n. | bez šumu | GSM | mic | SNR0 | SNR10 | vše |
| bez šumu | 83.87 | 82.90 | 83.87 | 22.26 | 51.77 | 60.00 |
| GSM | 85.00 | 86.13 | 84.84 | 24.52 | 44.19 | 59.88 |
| mic | 83.55 | 82.58 | 83.55 | 20.97 | 50.00 | 59.10 |
| SNR0 | 34.03 | 29.68 | 34.03 | 54.19 | 64.52 | 45.39 |
| SNR10 | 53.39 | 49.68 | 53.39 | 54.03 | 77.42 | 58.28 |
| vše | 74.03 | 75.00 | 74.03 | 45.16 | 64.84 | 64.45 |

Tabulka 8.4: Úspěšnost rozpoznávání [%] pro různé kombinace zkreslených dat použitých jako trénovací nebo testovací množina

| Hodnoty $dv$ | | | | | | |
|---|---|---|---|---|---|---|
| | rozpoznávané nahrávky | | | | | |
| trénovací n. | bez šumu | GSM | mic | SNR0 | SNR10 | vše |
| bez šumu | 0.995 | 0.992 | 0.995 | 1.024 | 0.961 | 1.039 |
| GSM | 1.008 | 0.994 | 1.008 | 1.034 | 0.956 | 1.033 |
| mic | 0.994 | 0.990 | 0.994 | 1.024 | 0.963 | 1.041 |
| SNR0 | 1.131 | 1.138 | 1.131 | 0.993 | 0.977 | 1.097 |
| SNR10 | 1.134 | 1.146 | 1.134 | 1.063 | 0.993 | 1.135 |
| vše | 0.984 | 0.973 | 0.984 | 0.963 | 0.929 | 0.991 |

Tabulka 8.5: Srovnání variancí testovacích dat a modelu pomocí hodnot $dv$ pro různé kombinace zkreslených dat použitých jako trénovací nebo testovací množina

| Hodnoty $dm$ | | | | | | |
|---|---|---|---|---|---|---|
| | rozpoznávané nahrávky | | | | | |
| trénovací n. | bez šumu | GSM | mic | SNR0 | SNR10 | vše |
| bez šumu | 0.026 | 0.173 | 0.026 | 0.759 | 0.644 | 0.282 |
| GSM | 0.178 | 0.051 | 0.177 | 0.822 | 0.711 | 0.346 |
| mic | 0.034 | 0.173 | 0.034 | 0.761 | 0.647 | 0.284 |
| SNR0 | 0.695 | 0.747 | 0.695 | 0.041 | 0.290 | 0.389 |
| SNR10 | 0.669 | 0.748 | 0.669 | 0.302 | 0.025 | 0.346 |
| vše | 0.271 | 0.310 | 0.271 | 0.427 | 0.348 | 0.037 |

Tabulka 8.6: Srovnání středních hodnot testovacích dat a modelu pomocí hodnot $dm$ pro různé kombinace zkreslených dat použitých jako trénovací nebo testovací množina

# 9 Závěr

Cílem této práce byla implementace a optimalizace vybraných metod rozpoznávání řeči z hlediska času výpočtu a také analýza a optimalizace z hlediska úspěšnosti rozpoznávání.

Úloha rozpoznávání řeči se skládala ze čtyř základních částí - parametrizace, akustického modelování, jazykového modelování a dekódování. Parametrizace signálu byla provedena pomocí metody melovských kepstrálních koeficientů. Akustické modelování bylo realizováno s využitím skrytých Markovových modelů (HMM).

Optimalizace z hlediska času výpočtu byla zaměřena na zrychlení výpočtu akustických pravděpodobností a dekódování jako celku. Pro snížení časové náročnosti dekódování bylo využito prořezávání. Během výpočtu algoritmu forward-backward byly v matici hodnot $\alpha$ ponechávány pouze některé hodnoty, odpovídající několika „nejlepším cestám“. Tím se snížilo celkové množství operací, které bylo třeba během výpočtu provádět.

Výpočet akustických pravděpodobností byl převeden na kombinaci dvou jednodušších operací: násobení matic a funkce *addLog*, představující přirozený logaritmus součtu dvou hodnot, které máme v dispozici v logaritmech. Pro rychlejší výpočet násobení matic lze použít specializované knihovny, funkce *addLog* pak byla zrychlena využitím aproximace Taylorovým rozvojem a instrukční sady SSE, která umožňuje provádět některé operace s více hodnotami najednou. V závislosti na požadované přesnosti bylo dosaženo až 75% zrychlení výpočtu této funkce.

Zrychlení výpočtu akustických pravděpodobností hraje velkou roli zejména při rozpoznávání řeči v reálném čase. Vysoká výpočetní náročnost úlohy rozpoznávání zde často vyžaduje kompromisy mezi rychlostí a přesností. Její snížení tak umožní získání lepších výsledků.

Druhá část práce se zabývala analýzou a optimalizací rozpoznávání řeči z hlediska jeho úspěšnosti. K tomuto účelu byla zvolena jednoduchá úloha, získané poznatky však lze využít i v úlohách složitějších.

Pro trénování modelu a následné rozpoznávání promluv byla využívána množina 62 telefonních nahrávek, ve kterých řečníci vyslovovali číslovky od nuly do devíti v různém pořadí.

Úspěšnost rozpoznávání byla počítána s využitím Levenshteinovy vzdálenosti, pro detailnější vyhodnocení sloužil program HResults z balíku nástrojů HTK.

Jednotlivá slova byla modelována jako celky, prvním úkolem při optimalizaci úspěšnosti rozpoznávání pak bylo stanovení vhodného počtu stavů HMM připadajících na každé z nich. Podle úspěšnosti rozpoznávání se jako nejlepší ukázalo 5 nebo 9 stavů modelu na slovo. Jelikož vyšší počet stavů znamená delší dobu výpočtu, byla zvolena nižší z těchto hodnot, tedy 5 stavů na slovo.

Dále bylo třeba stanovit, zda a jakým způsobem normalizovat příznakové vektory nahrávek. Byly srovnávány úspěšnosti rozpoznávání pro tři možnosti: výpočet bez normalizace, normalizace dat na nulovou střední hodnotu, nebo normalizace na nulovou střední hodnotu a jednotkovou maximální velikost. Jako nejlepší se ukázala poslední z těchto variant.

Následně byla pro dekódování aplikována penalizace slov. Jejím účelem bylo redukovat počet rozpoznaných slov, jelikož dekodér měl tendenci upřednostňovat větší množství kratších slov před menším množstvím slov delších. Byl hledán vhodný způsob určení optimální velikosti penalty, který by nevyžadoval provedení rozpoznávání pro příliš velké množství různých hodnot. Nakonec bylo rozhodnuto velikost penalty stanovovat na základě celkového počtu rozpoznaných slov, a to tak, aby byl roven skutečnému počtu slov v promluvách. Závislost počtu slov na penaltě se ukázala být monotónní funkcí, je tedy možné optimální hodnotu nalézt interpolací na základě pouze několika hodnot.

Poslední částí práce byla analýza výsledků rozpoznávání. Ta byla provedena na úrovni jednotlivých stavů HMM. Pro srovnání byly nahrávky rozpoznávány rovněž se znalostí referenčního přepisu a na základě takto získané referenční posloupnosti stavů byly hledány nejčastěji se vyskytující záměny. Bylo zjištěno, že velmi často dochází k záměně slova „dva“ za slovo „nula“ a slova „pět“ za „devět“ a naopak. Příčinou je značná podobnost těchto dvojic slov v mluvené formě.

Následně byl zkoumán vliv zkreslení dat na kvalitu rozpoznávání. Nejprve k tomuto účelu byly použity původní nahrávky, jejichž příznakové vektory byly zkresleny přidáním šumu. Jednalo se o náhodný šum s nulovou střední hodnotou a různou variancí, a dále naopak o posun hodnot o konstantu. Později byly rovněž zkresleny přímo samotné nahrávky, a to aplikací ztrátové komprese kodekem GSM 06.10, dále simulací charakteristiky stolního mikrofonu, a přidáním šumu z automobilu. Rozpoznávání řeči pak bylo prováděno pro různé kombinace zkreslení trénovací a testovací množiny nahrávek. Zejména šum z automobilu velmi snížil výslednou úspěšnost rozpoznávání.

Vedle samotné úspěšnosti rozpoznávání byly rovněž sledovány parametry $dm$ a $dv$, které byly zavedeny jako vyjádření rozdílů ve středních hodnotách a variancích mezi modelem a odpovídajícími daty pro stejné stavy. Pro správnou funkci dekodéru by tyto rozdíly měly být minimální. Při použití zašuměných nahrávek však vykazovaly odpovídající změny. Analýza středních hodnot a variancí umožňuje zjistit, zda byl akustický model správně natrénován a jakým nejvhodnějším způsobem případně zvýšit úspěšnost rozpoznávání.

# Literatura


[1] BĚHUNEK, M. *Rozpoznávání řeči při různé kvalitě vstupního signálu.* Diplomová práce, Praha: ČVUT FEL, 2010

[2] FORNEY, G. D., Jr. The Viterbi Algorithm. *Proceedings of the IEEE*, 1973, roč. 61, č. 9, s. 268-278, ISSN 0018-9219

[3] GIBBON, D., MOORE, R. K., WINSKI, R. *Handbook of standards and resources for spoken language systems.* Berlin: Walter de Gruyter, 1997, ISBN 978-3110153668

[4] HERMANSKY, H. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 1990, roč. 87, č. 4, s. 1738-1752, ISSN 0001-4966

[5] KHREICH, W. et al. On the memory complexity of the forward-backward algorithm. *Pattern Recognition Letters*, 2010, roč. 31, č. 2, s. 91-99, ISSN 0167-8655

[6] KUNEŠOVÁ, M. *Optimalizace algoritmů pro zpracování řečového signálu.* Bakalářská práce, Plzeň: ZČU FAV, 2011

[7] LIU, Y. et al., Accelerate Acoustic Likelihood Computations on GPU for Speech Recognition. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)* [online], 2013, ISBN 978-90-78677-61-1, s. 1019-1022 [cit. 2013-04-27] Dostupné na www: <http://www.atlantis-press.com/php/download_paper.php?id=4687>

[8] PSUTKA, J., MÜLLER, L. a kol. *Mluvíme s počítačem česky.* Praha: Academia, 2006, ISBN 80-200-1309-1

[9] RABINER, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 1989, roč. 77, č. 2, s. 257-286, ISSN 0018-9219

[10] SZÖKE, I. Jak se počítač učí rozpoznávat mluvenou řeč. *OSEL* [online]. 2010 [cit. 2013-04-14] Dostupné na www: <http://www.osel.cz/index.php?clanek=5152>

[11] TUŠER, P. *Nastavení MFCC parametrizace v úloze rozpoznávání řečníka.* Diplomová práce, Plzeň: ZČU FAV, 2006

[12] YOUNG, S. et al. *The HTK Book (for HTK Version 3.4)*. [online] Cambridge University Engineering Department, 2006, [cit. 2013-04-26] Dostupné na www: <http://htk.eng.cam.ac.uk/docs/docs.shtml>

[13] ZAJÍC, Z. *Adaptace akustického modelu v úloze s malým množstvím adaptačních dat*. Disertační práce, Plzeň: ZČU FAV, 2012