Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DOPLOMOVÁ PRÁCE

PLZEŇ, 2015 **MILAN JAROLÍN**

P R O H L Á Š E N Í

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

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

v Plzni dne 1. března 2015 .....

vlastnoruční podpis

P O D Ě K O V Á N Í

Tímto bych chtěl poděkovat vedoucímu diplomové práce Ing. Mgr. Josefu Psutkovi, Ph.D. za velmi užitečnou pomoc a cenné rady při zpracování práce.

Anotace

V této diplomové práci se zabývám problematikou návrhu robustního detektoru řeči a ověření jeho funkce v různých akustických prostředích.

Při návrhu vycházím z upraveného statistického přístupu pro návrh detektorů řeči, při němž neuvažuji detekovanou řeč jako celek ale dělím ji na menší jednotky, fóny, potažmo zvuky pro šum. Pro reprezentaci těchto jednotek používám skryté Markovské modely.

Dále pomocí experimentů popisuji použitý postup při nastavení parametrů detektoru řeči pro různá akustická prostředí a rozhodovací pravidla implementovaného klasifikátoru. Ve výsledku pak zkoumám vliv volby těchto parametrů a modifikací rozhodovacího pravidla na kvalitu navrženého detektoru v porovnání s referenční hodnotou na základě vhodně zvolených kritérií.

Klíčová slova

detektor řeči, VAD, statistický přístup, MFCC, akustické prostředí, akustický model, Markovský model, HMM

Anotation

In this thesis I deal with the issue of robust speech detector design and followed verification of its function in various acoustic environments.

The design is based on modified statistical approach to design speech detectors, in which the detected speech is not considered as a whole but divided into smaller units, phones or sounds for noise signals. To represent these units using hidden Markov models.

Furthermore, through experiments I describe the procedure used in setting the parameters of the speech detector in various acoustic environments and different decision rules implemented in classifier. As a result, examine the influence of the modification of these parameters and a decision rule in comparison with reference value on the basis of appropriately selected criteria.

Key words

speech detector, VAD, statistical approach, MFCC, acoustic environment, acoustic model, Markov model, HMM

Obsah

Reference

[1] Psutka J., Matoušek J., Muller Z., Radová V. *Mluvíme s počítačem česky.* Vyd. 1. Praha: Academia, 2006, 746 s. Česká matice technická, roč. 111, č. spisu 502. ISBN 8020013091.

[2] Peinado A., Segura J. *Speech Recognition Over Digital Channels—Robustness and Standards.* West Sussex, U.K.: Wiley, 2006 str. 274. ISBN: 978-0-470-02400-3.

[3] Sohn J., Kim N.S., Sung W. *A statistical model-based voice activity detection. IEEE Signal Processing Letters* [online]. 1999, vol. 6, issue 1, s. 1-12 [cit. 2015-03-17]. DOI: 10.2174/978160805172411101010001.

[4] Srinivasan K., Gersho A., Cheung M.T. *Voice activity detection for cellular networks Proc. of IEEE Workshop on Speech Coding for Telecommunications* [online]. 1993 [cit. 2015-03-17]. DOI: 10.14711/thesis-b622454.

[5] Ramirez J., Segura J.C., Benitez C., de la Torre A., Rubio A. *Efficient voice activity detection algorithms using long-term speech information. Speech Communication* [online]. 2004, vol. 42, 3-4, s. 271-287 [cit. 2015-03-17]. DOI: 10.1016/j.specom.2003.10.002.

[6] Ramirez J., Segura J.C., Benitez C., Garcia L., Rubio A. *Statistical voice activity detection using a multiple observation likelihood ratio test. IEEE Signal Processing Letters* [online]. 2005, vol. 12, issue 10, s. 689-692 [cit. 2015-03-17]. DOI: 10.1109/lsp.2005.855551.

[7] Tanyer S.G., Ozer H. *Voice activity detection in nonstationary noise. IEEE Transactions on Speech and Audio Processing* [online]. 2000, vol. 8, issue 4, s. 478-482 [cit. 2015-03-17]. DOI: 10.1109/89.848229.

[8] Freeman D. K., Cosier G., Southcott C. B., Boyd I. *The voice activity detector for the Pan-European digital cellular mobile telephone service. International Conference on Acoustics, Speech, and Signal Processing* [online]. 1989 [cit. 2015-03-17]. DOI: 10.1787/340460741623.

[9] Jarolín M. *Optimalizace rozmístění pásmových filtrů v MFCC s ohledem na zpracovávanou množinu řečníků*,
`http://hdl.handle.net/11025/5845`

[10] http://www.freesound.org/

[11] http://www1.icsi.berkeley.edu/Speech/docs/HTKBook3.2.pdf

[12] http://dsp.stackexchange.com/questions/15938/is-this-a-correct-interpretation-of-the-dct-step-in-mfcc-calculation

[13] http://sox.sourceforge.net/Docs/FAQ

1 Úvod

I přes to, že problém detekce řeči čili VAD (Voice Active Detection) není nový a při jeho řešení během minulých dekád množství výzkumníků zkoumalo odlišné strategie pro detekci řeči v zašuměném prostředí a následný vliv VAD na systémy zpracování řeči, určitě jej nemůžeme ještě považovat za vyřešený. Koneckonců, problém vysoce přesné detekce řeči je srovnatelný s problémem rozpoznávání řeči.

Spolu se stále rozvíjejícími se technologiemi, rostoucím významem oblasti zpracování řeči a poptávkou po kvalitních aplikacích, se stává role VAD čím dál důležitější. Rozmanitost a různá povaha řeči a šumu v pozadí to ale komplikují.

Klasické metody dnes dokáží pracovat s uspokojivou přesností ale pouze za pro ně předem známých podmínek. Bohužel tyto algoritmy mají problémy v prostředích s velmi nízkými hodnotami poměru mezi řečí a šumem tzv. SNR (Signal to Noise Ratio), obzvláště pokud je šum v pozadí nestacionární. V takových situacích je pro ně nemožné rozlišit od sebe řeč a šum použitím jednoduché techniky prahové detekce, když části řečové promluvy jsou překryty výrazným šumem pozadí [8].

V současnosti je proto aktuální snaha o konstrukci vysoce robustních detektorů řeči, které budou podávat stabilní výsledky pro různá akustická a to i vysoce zašuměná prostředí.

Detektor řeči, který v této práci navrhuji, bude primárně sloužit pro úlohu tzv. „frame-dropping“, neboli zahazování rámců, v procesu automatického titulkování televizních pořadů a to ve smyslu detekce úseků nahrávek kdy má být tento proces aktivní.

V mém případě se pokouším přistoupit k problému detekce řeči z jiného hlediska a neuvažovat řeč ani šum jako celek ale rozdělit je na jemněji specifikované části. Pro řeč se přímo nabízí rozdělení řeči na jednotlivé fonémy. Naproti tomu šum už tak intuitivně rozdělit nejde. Jeho definice je příliš obecná. Proto jsem se při dělení šumu omezil a definoval jeho části jako možné zvuky vyskytující se ve zkoumaných prostředích čili v kanceláři, na ulici a při sportovních utkáních.

2 Teoretická analýza detekce řeči

Tato práce, jak už název napovídá, primárně zkoumá a rozvádí, pro specifikovaná akustická prostředí, problematiku detekce řečového signálu, což je proces, jehož primární funkcí je poskytovat údaj o výskytu řeči pro usnadnění jejího zpracování stejně tak jako možného poskytnutí časového označení přechodů pro začátek a konec řečového segmentu ve zvukovém signálu ať už se zpracovává v reálném čase nebo hromadně z nahraných promluv.

Samotný VAD je jednou ze zásadních otázek v tématu zpracování řeči, má široké využití v oboru řečově zaměřených aplikacích, včetně úloh zpracování řeči v reálném čase nebo parametrizace a robustního rozpoznávání řečového signálu. V těchto oblastech je kriticky důležitý pro dosažení vysoké úrovně výkonnosti.

Zpracování v reálném čase má význam pro široký okruh reálných implementací například při hands-free telefonování, uchovávání řeči nebo pro digitální mobilní rádio. Výhodou může být nižší průměrná spotřeba energie v mobilních zařízeních nebo vyšší průměrná přenosová rychlost pro simultánní služby jako přenos dat.

V robustním rozpoznávání řeči, hraje VAD důležitou roli ve dvou hlavních aplikacích. První je odhad statistik šumu v pozadí potřebných pro algoritmy potlačení šumu jednokanálového signálu, jako je spektrální subtrakce nebo Wienerova filtrace (WF). Ačkoli bylo navrženo několik technik pro kontinuální aktualizace odhadu šumu v pozadí, obvykle se vypočítává během neřečových period, a proto potřebujeme algoritmus VAD. Tyto techniky odhadují spektrum šumu během neřečových period ve smyslu kompenzace pro jeho škodlivý vliv na řečový signál.

Druhé použití spočívá ve vyřazení neřečových úseků jako část předzpracování dat pro rozpoznávání řeči. To se obecně nazývá „zahazování rámců“. Odstraněním neřečových pasáží ze vstupního datového toku systému rozpoznávání řeči efektivně snižuje četnost chyby vložením v systému. Obecněji, snaží se zabránit situacím kdy i přes neřečový signál na vstupu systému pro rozpoznávání řeči je výstupem klasifikátoru náhodný řečový prvek.

Pro tyto účely, bylo navrženo množství VAD algoritmů, které se snaží vyvážit kritéria jako je zpoždění, citlivost, přesnost a výpočetní náročnost. Nicméně, zlepšení záleží hlavně na procentuálním výskytu pauz během řeči a spolehlivosti VAD použitého pro detekci těchto intervalů. Na jednu stranu, je samozřejmě výhodné mít nízkou procentuální hlasovou aktivitu, ale na druhou stranu, bychom se měli vyvarovat odříznutí řečových úseků pro zachování informace a kvality. Toto je klíčový problém pro VAD algoritmus zejména za obtížných podmínek šumu. VAD je tedy zásadnější pro nestacionární šumové prostředí, protože je potřeba k aktualizacím statistik neustále

měnícího se šumu a má tak vliv na chybu špatnou klasifikací, která silně ovlivňuje výkon systému [2].

2.1 Reprezentace dat po mikrosegmentech

Jelikož existuje nepřeberné množství přístupů, které vycházejí z různých předpokladů, neexistují nijak striktně definovaná pravidla či omezení pro vstupy do algoritmu detekce řeči. Aby tedy bylo možné detekovat řeč v nějakém časovém měřítku, musíme si jej nejdříve nadefinovat a tomu přizpůsobit reprezentaci dat pro další zpracování. Při definici musíme uvážit dvě hlediska, potřebná časová přesnost detekce řeči a dynamika řeči a šumu. Přestože je lidská řeč komplexní, souvisle časově proměnný, vysoce dynamický proces, je na rozdíl od šumu omezena lidským hlasovým ústrojím a proto můžeme parametry matematického modelu, který jej popisuje, považovat, v časovém intervalu v řádu milisekund, za konstantní. Jelikož pro naprostou většinu aplikací je přesnost v řádu několika milisekund dostačující můžeme definovat velikost intervalu pro zpracování vstupních dat podle tohoto kritéria. Tyto intervaly se nazývají mikrosegmenty nebo rámce. Přesnou velikost je možné zvolit individuálně, doporučuje se ji ale volit v rozmezí $10-35[ms]$ v závislosti na dalších krocích. Pokud je zvolena nižší zvyšuje se „hrbolatost“, opačně při vyšších hodnotách klesá přesnost klasifikace. Pro účely VAD je možné dále pracovat s takto rozdělenými daty. Pro přístupy, které však nejsou založené na výpočtu energie, je běžné dříve aplikovat jednu z parametrizačních metod z množiny metod krátkodobé analýzy, jako je MFCC, LPC, PLP atd. Rozdělení dat na rámce nemusí být nutně součástí VAD algoritmu, je samozřejmě možné si data předpřipravit externím nástrojem a upravit podle toho načítání vstupu. Přístup zpracování audio nahrávek po rámcích je běžně používán, mimo jiné, i v problematice rozpoznávání řeči.

2.2 Popis jednotlivých kroků VAD

Obecných průběh VAD algoritmu můžeme v zásadě rozdělit do třech důležitých kroků, ve kterých se volí způsob, jakým bude algoritmus k detekci řeči přistupovat. Konkrétní implementace se však mohou i přes stejný přístup v detailech lišit.

- Výpočetní metrika
  Volba metriky je zcela zásadní pro funkci a výkon detektoru řeči. Je to postup jehož smyslem je z každého rámce zpracovávané promluvy extrahovat kvantitativní parametr nebo vektor parametrů, který popisuje jeho charakteristiky podle zvoleného přístupu. Například to může být energie, pravděpodobnost, střední počet průchodů nulou, MFCC koeficienty atp.

- Rozhodovací pravidlo
  Detektor řeči je vlastně jednoduchý klasifikátor, který podle vstupních parametrů klasifikuje jednotlivé rámce do dvou tříd představujících řeč a šum. Rozhodovací pravidlo má v této analogii povahu klasifikační funkce, která vstup v podobě metriky porovná s buď to fixní, nebo adaptivní prahovou hodnotou a rozhodne o příslušnosti zpracovávaného rámce. Práh se nastaví buď uživatelem, nebo se jednorázově vypočte na začátku promluvy z prvních n rámců, za heuristického předpokladu výskytu ticha na začátku každé promluvy. Hodnota adaptivního prahu je obvykle aktualizována ze segmentů klasifikovaných jako neřečové.

- Vyhlazení výstupu
  Při detekci řeči po jednotlivých rámcích, obzvláště v prostředích s nestacionárním šumem nebo s nízkým SNR, může na výstupu docházet k chybám špatnou klasifikací pro obtížně vyhodnotitelné úseky v náhodných segmentech. Pokud uvážíme fakt, že přechody mezi řečí a tichem se vyskytují v intervalech mnohem delších než je velikost jednoho mikrosegmentu, měla by se pro výslednou klasifikaci rámce využít i informace o prvotním rozhodnutí ze sousedních rámců. Ve výsledku je klasifikace rámců mnohem plynulejší a přirozenější. Pokud budeme chtít při vyhlazení uvažovat okolní rámce na obě strany od aktuálního, tak při zpracování v reálném čase, se do procesu detekce řeči vnáší zpoždění vyhodnocení právě o velikosti poloměru okolí.

2.3 Klasický přístup VAD

Základní funkcí VAD algoritmu založeném klasickém přístupu je extrakce určitých měřitelných funkcí a kvantitativních vlastností ze vstupního signálu a porovnání těchto hodnot s prahovou hodnotou, obvykle získanou z charakteristik šumu a řečového signálu. Rozhodnutí o řečové aktivitě je dáno, pokud měřené hodnoty překročí tento práh. V prostředí s nestacionárním šumem VAD vyžaduje časově proměnnou prahovou hodnotu [7].

Výkon těchto metod je přes jejich jednoduchost, v prostředích se SNR nad 10$[dB]$ uspokojivě vysoký i proto jsou stále tolik rozšířeny. Ve vysoce zarušených prostředích se SNR pohybujícím se okolo nuly či dokonce záporným však úspěšnost těchto metod rapidně klesá a stávají se nepoužitelnými. Různé druhy metrik byly navrženy pro design řešení problému VAD. Ty nejčastěji používanými jsou založené na měření celopásmové nebo subpásmové energie, počtu průchodu nulou nebo šikmosti a špičatosti neboli HOS (High Order Statistics).

2.3.1 Detektor řeči založený na celopásmové energii

Nejjednodušší přístup pro realizaci VAD je založený na předpokladu, že řečové úseky promluv mají významně vyšší celopásmovou energii než úseky obsahující pouze šum v pozadí. Vzhledem k odhadu průměrného logaritmu energie v pozadí je aktuální rámec klasifikován jako řeč či neřeč jednoduchým výpočtem rozdílu mezi aktuálním logaritmem energie a odhadem pozadí. Rozhodnutí je provedeno na bázi porovnání rozdílu s fixní prahovou hodnotou. Tento přístup může být nazýván i jako klasifikační pravidlo založené na odhadu okamžitého celopásmového SNR.

Zlepšení tohoto základního přístupu bylo navrženo v [4] rozšířením analýzy na několik spektrální pásem a použitím adaptivní prahové hodnoty pro prostředí s nestacionárním šumem. Výpočet hodnoty pro porovnání s prahem může mít pak podobu aritmetického či váženého průměru rozdílů pro jednotlivá pásma z vektoru, kde každý prvek má význam logaritmu energie signálu pro určité frekvenční pásmo.

Pro kompletní představu zde bude popsán jednoduchý proces detekce řeči pro dávkové použití založený na celopásmové energii s fixním prahem, průběžným odhadem parametrů pozadí a vyhlazením výstupu klasifikátoru.

- Výpočet logaritmu energie rámců:
  Nejdříve je nutné vypočíst zmíněný logaritmus energie pro každý zpracovávaný rámec $x(t)$

$$E(t) = 10log_{10}\frac{1}{N}\sum_{n=1}^{N} x(t,n)^2, \tag{1}$$

  kde $x(t,n)$ je n-tý vzorek aktuálního rámce, t je index pořadí rámce v promluvě a N je počet vzorků v rámci.

- Odhad logaritmu energie šumu v pozadí:
  Abychom získali kýženou prahovou hodnotu pro klasifikaci, uvažuje se počáteční část každé promluvy (10 - 20 rámců) jako neřečová a z obsažených rámců lze inicializovat počáteční odhad úrovně šumu v pozadí výpočtem aritmetického průměru logaritmu energie.

$$\theta_e = \frac{1}{T}\sum_{t=1}^{T} E(t), \tag{2}$$

  kde θ_e je hodnota odhadu v čase $t = 1$ a T je počet rámců na začátku promluvy označených jako inicializační úsek. Pro rámce s $t > T$ označené VAD algoritmem jako neřečové se hodnota odhadu θ_e aktualizuje implementací rekurzivního filtru 1. řádu, sledujícího pomalé variace energie šumu v pozadí.

$$\theta_e = \alpha\theta_e + (1-\alpha)E(t) \tag{3}$$

  Typické hodnoty pro α jsou v rozmezí $(0,95 - 0,99)$.

- Klasifikace:
  Pro vyhodnocení klasifikace chybí poslední informace a tou je hodnota prahu η_e. V tomto příkladě bude zadána fixně uživatelem a nebude se v průběhu detekce řeči v aktuální promluvě měnit. Klasifikátor má tak podobu

$$C(t) = \begin{cases} 1 & \text{pro } E(t) - \theta_e > \eta_e, \\ 0 & \text{pro ostatní.} \end{cases} \tag{4}$$

  Průběžné aktualizace prahu lze například dosáhnout v kombinaci s předchozím krokem, kdy by se hodnota prahu měnila v závislosti na vývoji odhadu šumu v pozadí a $max_{\forall t}\{E(t)\}$.

- Vyhlazení výstupu klasifikace:
  Předchozím krokem by detektor řeči mohl končit, pro zajištění vyšší přesnosti je však doporučeno včlenit ještě další krok založený na pozorování toho, že přechod z neřečového do řečového úseku promluvy je běžně charakterizován rapidním nárůstem energie signálu. Naproti tomu je často pozorován pozvolný pokles v přechodech z úseků obsahujících řeč do těch co ji neobsahují, což může ve výsledku způsobovat chybu špatnou klasifikací. Tento problém se obvykle řeší tak, že VAD algoritmus zpožďuje rozhodnutí o přechodu z řečového do neřečového úseku. Tato technika se z angličtiny nazývá Hangover, volně přeloženo jako pozůstatek, a používá jednoduchý čítač. Příklad zdrojového kódu v programovacím jazyce Matlab, který popisuje funkci Hangover je přiložen níže.

```
prah_h = 10;
prah_r = 3;
n_h = 0;
n_r = 0;
for t = 1 : T
   if(C(t) == 1)
      C_f(t) = 1;
      n_r++;
      if(n_r > prah_r)
         n_h = prah_h;
      end
   else
      if(n_h == 0)
         C_f(t) = 0;
      else
         C_f(t) = 1;
         n_h--;
      end
   end
end
```

Na začátku je čítač n_h nastaven na nulu. Když $prah_r$ po sobě jdoucích rámců je klasifikováno jako řeč $C(t) = 1$, tak i finální klasifikace bude odpovídat řeči $C_f(t) = 1$ a čítač n_h se nastaví na hodnotu $prah_h$, což je počet rámců o které má být zpožděn přechod z řečového úseku do neřečového. Pokud je aktuální rámec původně klasifikován jako neřeč $C(t) = 0$, dekrementuje se hodnota čítače n_h. Rámec je finálně klasifikován jako neřeč $C_f(t) = 0$ až v případě kdy čítač $n_h = 0$.

2.4 Metody se statistickým přístupem

Tradiční VAD algoritmy jsou obvykle utvářeny užitím heuristického přístupu, což stěžuje optimalizaci relevantních parametrů. Se vzrůstajícím tlakem na robustnost systémů detekce řeči se začali zkoumat možnosti jiného pojetí problematiky než pouhou prahovou detekcí jak je tomu u klasických metod. Jedním z odlišných a v současnosti častěji používaných přístupů k návrhu VAD se snahou jej optimalizovat je formulovat rozhodnutí ze statistického hlediska.

2.4.1 Detektor řeči založený na statistickém modelu

Předpokládaje statistický model pro řečový signál byl v [3] navržen VAD algoritmus založený na LRT testu (Likelyhood Ratio Test).

Pro řeč narušenou aditivním nekorelovaným šumem, vytvoříme dvě hypotézy:

$$H_0 : X = N \qquad \rightarrow \text{Neobsahující řeč} \tag{5}$$
$$H_1 : X = N + S \quad \rightarrow \text{Obsahující řeč} \tag{6}$$

kde S,N,X jsou L dimenzionální vektory diskrétní Fourierovi transformace DFT (Discrete Fourier Transformation) koeficientů časového rámce, představující řeč, šum a zašuměnou řeč. S_k,N_k a X_k jsou k-té prvky daného vektoru.

V běžném statistickém modelu pro DFT koeficienty jsou jednotlivé koeficienty pro každý proces uvažovány jako asymptoticky nezávislé Gaussovské náhodné proměnné. Pod tímto modelem jsou podmíněné hustoty pravděpodobnosti PDF (probability density function) pro hypotézy H_0 a H_1 definovány jako

$$p(X|H_0) = \prod_{k=0}^{L-1} \frac{1}{\pi\lambda_N(k)} exp\left\{-\frac{|X_k|^2}{\lambda_N(k)}\right\}, \tag{7}$$
$$p(X|H_1) = \prod_{k=0}^{L-1} \frac{1}{\pi\left[\lambda_N(k) + \lambda_S(k)\right]} exp\left\{-\frac{|X_k|^2}{\left[\lambda_N(k) + \lambda_S(k)\right]}\right\}, \tag{8}$$

kde $\lambda_N(k)$ a $\lambda_S(k)$ jsou variance N_k respektive S_k. Pravděpodobnostní míra

pro k-té frekvenční pásmo neboli DFT koeficient je proto

$$\Lambda = \frac{p(X|H_1)}{p(X|H_0)} = \prod_{k=0}^{L-1} \Lambda_k, \tag{9}$$

$$\Lambda_k = \frac{1}{1+\xi_k} exp\left\{-\frac{\gamma_k \xi_k}{1+\xi_k}\right\}, \tag{10}$$

$$\xi_k \cong \frac{\lambda_S(k)}{\lambda_N(k)}, \tag{11}$$

$$\gamma_k \cong \frac{|X_k|^2}{\lambda_N(k)}, \tag{12}$$

kde ξ_k a γ_k jsou apriori a aposteriori SNR k-tého frekvenčního pásma. Rozhodovací pravidlo je formulováno na základě geometrického průměru pravděpodobnostní míry jednotlivých frekvenčních pásem, což je v souladu s předpokladem, že Fourierovy koeficienty jsou asymptoticky nezávislé náhodné proměnné. Může to být převedeno do logaritmické formy

$$log\Lambda = \frac{1}{L}\sum_{k=0}^{L-1} log\Lambda_k = \frac{1}{L}\sum_{k=0}^{L-1}\left\{\frac{\gamma_k\xi_k}{1+\xi_k} - log(1+\xi_k)\right\} \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{13}$$

kde η je fixní hranice. Variance šumu $\lambda_N(k)$ v každém frekvenčním pásmu může být odhadnuta z neřečových period použitím stejného přístupu popsaného výše pro estimaci úrovně šumu v pozadí.
Pro odhad apriorního SNR ξ_k, můžeme použít dva přístupy.

- Nejjednodušší je použít ML (Maximum Likelihood) estimátor založený na aposteriorním SNR

  $$\hat{\xi}_k^{(ML)} = \gamma_k - 1. \tag{14}$$

  Použitím tohoto estimátoru, dostaneme rozhodovací pravidlo přímo příbuzné s diskrétní formou Itakura-Saito zkreslení

  $$log\hat{\Lambda}^{(ML)} = \frac{1}{L}\sum_{k=0}^{L-1}\gamma_k - log\lambda_k - 1 \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{15}$$

  Pro které je známo, že má pozitivně definitní magnitudu, což způsobuje, že tato forma rozhodovacího pravidla upřednostňuje H_1.

- Pro potlačení této nerovnováhy bylo v [3] navrhnuto použití DD (Difference in differences) přístupu pro estimaci apriorního SNR

$$\hat{\xi}_k^{(DD)} = \alpha \frac{\hat{A}_k^2(n-1)}{\lambda_N(k)} + (1-\alpha) max\{\gamma_k(n) - 1, 0\}, \tag{16}$$

kde n je index rámce a $\hat{A}_k(n-1)$ je MMSE (Minimum Mean Square Error) odhad spektrální magnitudy signálu předchozího rámce. Tento přístup přináší hladší odhad apriorního SNR a snižuje fluktuaci odhadované pravděpodobnosti při neřečových periodách.

2.5 Metody využívající dlouhodobou informaci

Většina realizací VAD vykonává klasifikaci na bázi znalostí extrahovaných z právě zpracovaného rámce, stejně jako předchozí zmíněné metody. Okamžitá pozorování jsou porovnávány s průměrným odhadem procesu šumu v pozadí a rozhodnutí se zakládá na fixní nebo adaptivní hranici. Nedávno bylo nastíněno několik metod jako v [5] či [6], které kombinují informaci získanou z více rámců namísto použití jednorámcového pozorování. Hlavním cílem v tomto postupu je vytvořením kombinovaného pozorování zmenšit varianci rozhodování oproti jednorámcovému pozorování, získávaje tak robustnější rozhodnutí.
V [5], rozhodovací pravidlo vychází z takzvaných LTSD (Long Term Spectral Divergence) čili z dlouhodobé spektrální odchylky, definovaná pod tzv. dlouhodobou spektrální obálkou LTSE (Long Term Spectral Envelope). Nechť $X(k,l)$ je k-tý DFT koeficient daného rámce l a LTSE řádu N pro k-té pásmo l-tého rámce je definován jako

$$LTSE_N(k,l) = \max_{-N \leq j \leq N} \{X(k,l+j)\}. \tag{17}$$

V podstatě se tak spektrální obálka $LTSE_N$ pro aktuální rámec tvoří výběrem maximální hodnoty k-tého DFT koeficientu ze souboru $2N+1$ rámců okolo aktuálního rámce, pro každý koeficient zvlášť. LTSD pro aktuální rámec je potom definováno jako

$$LTSD_N(l) = 10log_{10}\left(\frac{1}{NFFT}\sum_{k=0}^{NFFT-1}\frac{LTSE_N^2(k,l)}{N^2(k)}\right), \tag{18}$$

kde $NFFT$ je počet DFT koeficientů a $N^2(k)$ je průměrný odhad energie k-tého pásma šumu v pozadí. Ze vzorce vyplývá, že se zjednodušeně jedná o průměrnou odchylku jednotlivých DFT koeficientů od jejich průměrů, udávanou v decibelech. Pokud je N nastaveno na nulu, dlouhodobá informace není uvažována a LTSE se stává okamžitým měřením spektrální divergence mezi aktuálním rámcem a šumem v pozadí. Zvyšováním řádu N vede k hladšímu LTSD, které poskytuje lepší výsledky VAD.

Robustnost použití rozhodovacího pravidla založeném vícerámcovém pozorování je také využita v [6], kde statistický VAD popsaný výše je upraven pro použití testu pravděpodobnosti vícenásobného pozorování MO-LRT (Multiple Observation - Likelihood Ratio Test). Používaje shodnou notaci, jednoduché LRT pro daný rámec t je uveden jako

$$\Lambda_t \cong \frac{p(X_t|H_1)}{p(X_t|H_0)}. \tag{19}$$

Pokud namísto jednoho rámce uvažujeme soubor $2m+1$ rámců okolo aktuálně zpracovávaného $\{X_{t-m}, \ldots, X_t, \ldots, X_{t+m}\}$, můžeme MO-LRT(Multi Observation - Likelyhood Ratio Test) definovat jako

$$\Lambda_t \cong \frac{p(X_{t-m}, \ldots, X_t, \ldots, X_{t+m}|H_1)}{p(X_{t-m}, \ldots, X_t, \ldots, X_{t+m}|H_0)}. \tag{20}$$

Tento výraz zahrnuje spojenou pravděpodobnost pozorování, která může být zjednodušena za předpokladu statistické nezávislosti získávání

$$\Lambda_t^{MO} = \prod_{l=-m}^{m} \frac{p(X_{t+l}|H_1)}{p(X_{t+l}|H_0)}. \tag{21}$$

Nakonec, pod stejným statistickým modelem použitým v [3], můžeme stanovit logaritmus míry pravděpodobnosti

$$log\Lambda_t^{(MO)} = \frac{1}{L} \sum_{l=-m}^{m} \sum_{k=0}^{L-1} \left\{ \frac{\gamma_{t+l,k}\xi_{t+l,k}}{1+\xi_{t+l,k}} - log(1+\xi_{t+l,k}) \right\} \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{22}$$

která obsahuje nejen apriorní znalost $\xi_{t,k}$ a aposteriorní $\gamma_{t,k}$ SNR aktuálního rámce ale také hodnoty korespondující k m předcházejícím a m následujícím rámcům.
Z provedených testů v [2] je zřejmé, že čím vyšší řád m tím získáváme test s vyšší rozlišovací schopností. Třídy řeči a neřeči jsou lépe oddělené a klasifikační chyby jsou tak redukovány.

3 Data

V této kapitole jsou popsané veškeré použité nahrávky a to jak pro trénovací tak testovací účely.

Protože byly shromážděny z různých zdrojů lišila se i původní kvalita jednotlivých nahrávek. Ze všeho nejdříve tak bylo potřeba uvést je do jednotného stavu. K tomu mi posloužila volně dostupná utilita SoX (Sound eXchange), což je z příkazové řádky ovládaný nástroj pro vzájemnou konverzi široké škály formátů reprezentující audio soubory. Dále umožňuje dávkové zpracování nahrávek a použití množství efektů během konverze, podrobněji [13].

Jako uniformní formát nahrávek jsem stanovil WAV pro jeho bezeztrátovost, široké použití a snadnou manipulaci, konkrétně pak kódované lineárně pulzní modulací PCM. Vzorkovací frekvence je uzpůsobena telefonnímu pásmu $(0; 4)[kHz]$ na hodnotu $f_{vz} = 8[kHz]$. V tomto stavu jsou nahrávky připraveny k dalšímu zpracování.

3.1 Akustická prostředí

Protože je tato práce věnována konstrukci robustního detektoru řeči s aplikací na úlohy automatického titulkování televizních pořadů, je nutné aby navrhovaný VAD algoritmus byl schopný poskytovat stabilně kvalitní výsledky za možné přítomnosti obsáhlé skupiny velmi odlišných typů šumů v pozadí.

Abych mohl různé druhy akustických prostředí nasimulovat, musel jsem přistoupit ke zjednodušení reprezentace množiny možných šumů v situacích typických pro jmenované aplikace a zvolil rozdělení do třech kategorií, které by dohromady svými vlastnostmi měli dostatečně dobře popisovat množinu šumů.

3.1.1 Tiché prostředí - *T*

První kategorii tvoří uzavřená prostředí se stacionárními či pomalu se měnícími charakteristikami šumu v pozadí a vysokým SNR. Zastupuje zvuky charakteristické pro kancelářské či studiové činnosti. Taková prostředí jsou ideální pro úlohy zpracování řeči, protože obsahují minimum nežádoucího signálu a dobře pro ně fungují i jednoduché detektory řeči. Nadále budu tuto kategorii označovat zkratkou T ze slova „tiché“.

3.1.2 Středně zašuměné prostředí - *N*

Dále také nazývána jako „běžné“. Představuje podmnožinu prostředí v čase velmi variabilně se měnících prostředí skládajících se z vysoce dynamických šumů v pozadí se SNR pohybujícím se mezi středními a nízkými hodnotami. Příznačné pro rušné ulice nebo pro cestování v různých dopravních prostředcích (vlak, autobus, auto, kolo). Dále pod zkratkou *N* ze slova „normální“.

3.1.3 Vysoce zašuměná prostředí - *R*

Největší obtíž v procesu detekce řeči nastává v případech v prostředí s velmi nízkým SNR. Tato prostředí zpravidla obsahují nestacionární šumy s vysokou intenzitou. Například se může jednat o zvuky z prostředí továren, datacenter, koncertů či jiných masových akcí nebo sportovních utkání. Dále pod zkratkou *R* jako „rušné“.

3.2 Trénovací data

Pro natrénování akustického modelu tak aby obsahoval prvky představující řeč i výše popsané šumy v dostatečné míře jsem musel trénovací množinu rozdělit.

3.2.1 Trénovací data pro řeč

Jedná se o soubor telefonních záznamů v rozsahu 3255 nahrávek o celkové délce přes 7 hodin, obsahujících části úředních dokumentů diktované množinou 100 řečníků s minimálním šumem v pozadí ale o ne příliš vysoké subjektivní kvalitě. Tyto data mi byla poskytnuta vedoucím mé práce.

Smyslem jeho použití je natrénování akustického modelu řeči. Který uchovává informaci o jisté charakteristické podobě všech definovaných řečových elementů a jejž využívám k detekci řečových úseků v promluvách.

3.2.2 Trénovací data pro šum

V tomto případě jsou to nahrávky původně získané ze zdroje [10], který obsahuje a volně poskytuje velké množství audio nahrávek různé povahy mimo jiné i pro podobné účely. Vybíral jsem takové ukázky šumů, které co nejlépe popisovali vybraná akustická prostředí zmíněná výše. Získané a dále předzpracované šumy jsem tedy také rozdělil do tří skupin.

Kategorie	Počet[ks]	Délka[min]
T	59	33
N	13	18
R	27	21

Tabulka 1: Rozložení nahrávek šumů v trénovací sadě dle kategorií.

3.3 Testovací data

Tyto nahrávky zpracovávám dvěma způsoby. Jednak jako testovací data pro vyvíjený VAD algoritmus a jednak pro vytvoření referenčního údaje používaného pro vyhodnocení úspěšnosti procesu detekce řeči. Ten získám zarovnám testovacích dat, na nich natrénovaným referenčním akustickým modelem, na jehož základě poté vyhodnocuji výkon a přesnost mnou navrženého detektoru řeči.

Pro každé ze třech akustických prostředích uvažovaných v této práci obsahuje testovací sada dvě nahrávky, všechny samostatně o délce přibližně 10 minut.

Tento rozsah by měl být dostačující pro vyhodnocení úspěšnosti a kvality detekce řeči pro zkoumaná prostředí. Více nahrávek je využíváno k ověření výsledků vyhodnocení. Kde pvní nahrávka slouží pro nastavení parametrů detektoru a druhá pro ověření funkce.

Pro prostředí T jsou to nahrávky obsahující diktát znění blíže neurčeného zákona ve studiovém prostředí. Pro prostředí N představují testovací nahrávky část záznamu ze závodu v bialtlonu. I přes to, že jde o sportovní tématiku obsah nahrávky spíše odpovídá charakteristikám středně zašuměného prostředí. Nakonec jsem jako vhodné reprezentativní nahrávky kategorie prostředí označené R zvolil audio záznam sportovního komentáře hokejového zápasu.

4 Návrh řešení

Tato kapitola popisuje můj postup při návrhu detektoru řeči. Před přístoupením k samotnému návrhu je nutné si ujasnit kritéria popisující jeho účel. Od toho se dále odvíjí celá struktura práce počínaje volbou metod pro jednotlivé kroky VAD algoritmu po volbu trénovacích i testovacích nahrávek. Dále upřesním výčet těchto kritérií a rovnou je nadefinuji pro můj konkrétní případ.

- Cílová aplikace:
  Jelikož způsobů použití existuje vícero je reálná možnost, že bez ujasnění si cílové aplikace může dojít k situaci kdy neuváženě zvolený přístup či metoda implementovaná do VAD zlepšující funkčnost pro určitou množinu aplikací může být ve výsledku kontraproduktivní. Mnou navrhovaný detektor řeči bude simulovat použití v úlohách automatického titulkování televizních pořadů.

- Přístup k zpracování dat:
  Rozhodnutí mezi dávkovým zpracováním a zpracováním v reálném čase. Některé přístupy mohou být příliš výpočetně náročné nebo přímo způsobují časové zpoždění do klasifikace výstupu (přístupy využívající dlouhodobé informace) a proto mohou nevhodné pro použití v reálném čase. I přes to, že pro obě zmíněné aplikace je nutné v reálném nasazení zpracování v reálném čase, volím pro tuto práci dávkové zpracování. Zvolil jsem tak z praktických důvodů pro zjednodušení ověřování funkčnosti vyvinutého VAD algoritmu na reálných již nahraných a analyzovaných datech.

- Pracovní akustické prostředí:
  Jak již bylo zmíněno v kapitole 2, na volbě pracovního prostředí extrémně záleží. Obzvláště důležitá je tato informace při plánovaném nasazení VAD v prostředích s nízkým SNR či nestacionární šumem v pozadí. Například u stále hojně používaných klasických metod detekce řeči založených na principu prahové detekce je použití v prostředích s nízkým SNR naprosto nevhodné. V dnešní době je velká snaha o vybudování robustního detektoru řeči, který by podával stabilně kvalitní výsledky pro libovolné prostředí.

  Tato práce se snaží právě o navržení a otestování takového robustního detektoru řeči, který by pro vybrané kategorie prostředích, definovaných v kapitole 3.1, podával uspokojivé výsledky.

Pokud jsou tyto kritéria stanovena, mohu přejít k procesu samotného návrhu adekvátní konfigurace VAD algoritmu, který by se dal rozdělit na následující fáze

1. Volba reprezentace dat
2. Volba metriky
3. Volba rozhodovacího pravidla
4. Volba způsobu vyhlazení výsledků
5. Volba vyhodnocení.

4.1 MFCC koeficienty

Jako vhodnou reprezentaci veškerých zpracovávaných nahrávek jsem pro další kroky zvolil často užívané Melovské frekvenční kepstrální koeficienty MFCC, pro jejich kvalitní reprezentaci řečového signálu a vysokou nekorelovanost jednotlivých koeficientů. Parametrizační metoda MFCC při zpracování řečového signálu využívá jak časové tak frekvenční oblasti a patří do množiny metod krátkodobého zpracování, které vychází z předpokladu stálosti lidského hlasového ústrojí při vyslovování určité hlásky v krátkém časovém intervalu. Signál vydávaný řečníkem pro tuto polohu řečového ústrojí může být popsán vhodně zvolenými spektrálními charakteristikami. MFCC popisují krátkodobé kepstrum, přičemž se snaží respektování znalostí o citlivosti vnímání zvuku o různých frekvencích lidským uchem, blíže v [1].

Experimentální závislost subjektivního vnímání výšky zvuku, udávaná v $[mel]$ u člověka je možné matematicky popsat následujícím vztahem

$$f_m = 2595 \log_{10} \left(1 + \frac{f}{700}\right), \tag{23}$$

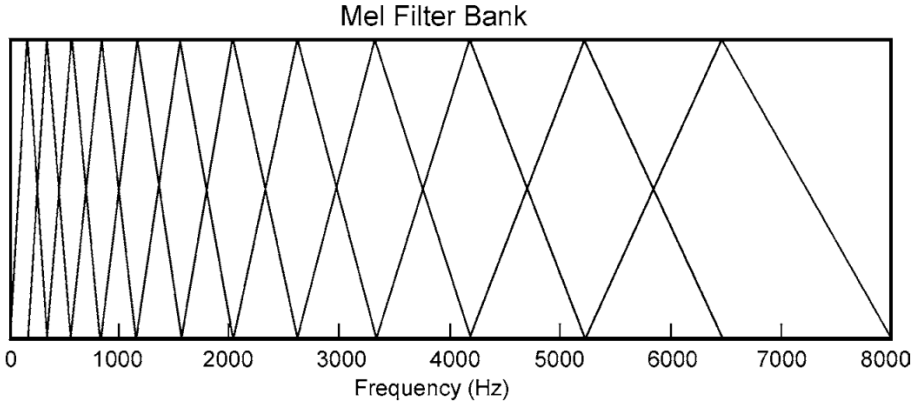
kde $f_m[mel]$ je frekvence v nelineární melovské škále a $f[Hz]$ je frekvence v původní lineární škále.

Další poznatek o chování lidského sluchu, kterého MFCC metoda využívá se týká takzvaných kritických pásem slyšení. Bylo dokázáno, že pokud zní dva tóny současně o frekvenci jen málo od sebe se lišící, lidské ucho je nedokáže od sebe odlišit. Kritická pásma jsou definována právě jako frekvenční intervaly s touto vlastností. Počet kritických pásem tedy závisí na přenášeném frekvenčním pásmu nahrávek.

Frekvence vzorkování F_v $[Hz]$	8000	16000	22050	44100
Přenášené pásmo $(0; B_w)$ $[Hz]$	(0;4000)	(0;8000)	(0;11025)	(0;22050)
Přenášené pásmo $(0; B_{mw})$ $[mel]$	(0;2146)	(0;2840)	(0;3174)	(0;3921)
Počet pásem $M*$	15	20	22	27

Tabulka 2: Typické hodnoty počtu filtrů $M*$ pro dané přenášené pásmo, [1]

Zjednodušeně pak výpočet MFCC pro každý rámec probíhá výpočtem frekvenčního spektra rámce $|S(f)|$ na které se aplikuje melovská banka trojúhelníkových vzájemně se překrývajících filtrů, jejichž počet a rozmístění odpovídá kritickým pásmům slyšení.

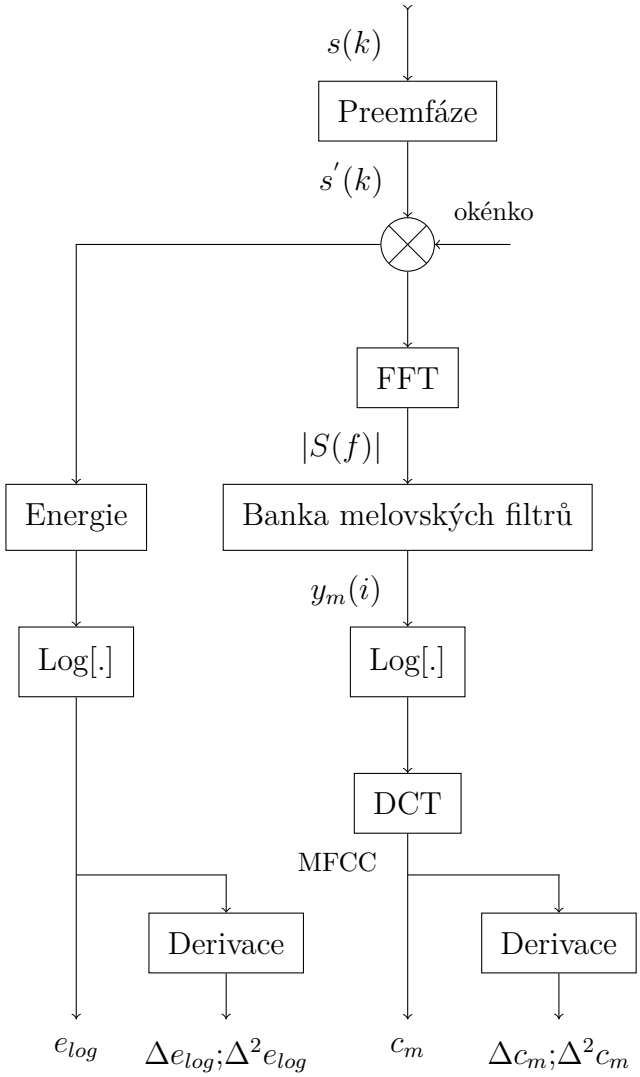


Obrázek 1: příklad rozmístění filtrů v Melovské bance filtrů pro přenášené pásmo $8kHz$, [12].

Naakumulované hodnoty filtrů y_m se převedou do logaritmického měřítka a z takto získaného vektoru se zpětnou diskrétní Fourierovou transformací získají výsledné koeficienty

$$c_m(j) = \sum_{i=1}^{M^*} \log y_m(i) cos\left(\frac{\pi j}{M^*}(i - 0.5)\right), \tag{24}$$

kde M^* je počet filtrů obsažených v bance melovských filtrů, $j = 0, 1, \cdots, M$ je index MFCC $c_m(j)$ a M označuje počet melovských kepstrálních koeficientů a obvykle se volí v rozmezí $10 \leq M \leq 13$. Detailněji je průběh výpočtu MFCC koeficientů s delta a akceleračními koeficienty je znázorněna na obrázku 2.



Obrázek 2: Schéma algoritmu MFCC, [9]

Nultý koeficient $c_m(0)$ odpovídá logaritmu energie signálu, spíše se ale nahrazuje logaritmem krátkodobé energie E_n, vypočtené přímo z časových vzorků signálu

$$c_m(0) = e_{log} = \log E_n = \log \sum_{k=-\infty}^{\infty} [s'(k) \cdot w(n-k)]^2. \tag{25}$$

Použitím rozšířeného popisu o diferenční (delta) a akcelerační (delta-delta) koeficienty se získá informace charakterizující nejen polohu řečového ústrojí ale i jeho dynamické vlastnosti. Finální dimenze vektoru MFCC pro jeden rámec je poté rovna $dim(c_m) = 3M$.

4.2 Upravený statistický přístup

Již v úvodu bylo zmíněno, že proces detekce řeči je si v mnohém podobný s procesem rozpoznání řeči. Z toho důvodu se při návrhu VAD algoritmu pokouším aplikovat postup používaný právě při rozpoznávání řeči zmíněný v [2].

Za předpokladu, že $W = w_1, w_2, \ldots, w_T$ je posloupnost klasifikačních stavů nabývajících hodnot z množiny $V = s, n$, kde s představuje řeč a n šum a nechť $X = x_1, x_2, \ldots, x_T$ je posloupnost vektorů příznaků MFCC, kde T je počet rámců v promluvě, je cílem nalézt posloupnost $\hat{W}$, tedy nejpravděpodobnější posloupnost hodnot pro danou akustickou informaci X. Klasifikační pravidlo pro každý rámec t je tak dáno Bayesovým vztahem

$$\hat{W}_t = arg \max_V P(V|X_t) = arg \max_V \frac{P(V)P(X_t|V)}{P(X_t)}, \tag{26}$$

kde $P(X_t|V)$ je podmíněná pravděpodobnost výskytu vektoru příznaků X_t, při známé informaci o klasifikaci rámce, $P(V)$ je apriorní pravděpodobnost výskytu řeči či šumu v promluvě a $P(X)$ je apriorní znalost výskytu konkrétního charakteristického vektoru. Protože $P(X)$ není funkcí V, lze ji při hledání maxima vypustit. Podobně při detekci řeči by bylo možné pracovat s pravděpodobností $P(V)$ pokud bych měl tuto informaci, obecně ale toto rozložení není známé a tak se nahrazuje rovnoměrným $P(V) = 0.5, 0.5$. V tomto případě se tak klasifikační pravidlo zjednodušilo na následující podobu

$$\hat{W}_t = arg \max_V P(V|X_t) = arg \max_V P(X_t|V). \tag{27}$$

Kde podmíněné rozdělení pravděpodobnosti $P(X_t|V)$ nese informaci o akustickém modelu a před zahájením procesu klasifikace je nutné určit tuto informaci, většinou na základě trénování z řečových dat.

4.2.1 Akustický model

Účelem akustického modelu v této úloze je poskytování co nejpřesnějšího odhadu podmíněné pravděpodobnosti $P(X_t|V)$ pro $\forall t$. „Akustické modely by měli být flexibilní, přesné a účinné“ [2]. Flexibilní musí být, protože podmínky za nichž je provozován detektor řeči, jsou často zcela odlišné od podmínek trénování (odlišné hlasy, odlišný způsob artikulace, odlišné tempo řeči, odlišné vlastnosti akustivého kanálu, odlišné akustické pozadí). Přesnost je naopak potřebná kvůli požadavku odlišit foneticky podobné úseky s lingvisticky odlišnými významy. Nakonec účinnost je nutná v případě kdy systém detekce řeči je nasazován v reálných aplikacích a odezva klasifikátoru musí být dostupná v reálném čase. Jako velmi efektivní se ukázalo být pro tuto

netriviální úlohu použití množiny tzv. skrytých Markovských modelů HMM na rozlišovací úrovni fonémů. Předpokládá se, že v diskrétních časových okamžicích je proces v jediném stavu. Problém nalezení co nejlepšího odhadu rozdělení pravděpodobnosti $P(X_t|V)$ je závislý na určení topologie HMM a typu hodnot jeho parametrů. Soubor těchto neznámých parametrů lze v zásadě určit dvojím způsobem. Pomocí apriorních znalostí vyberu vhodnou strukturu HMM, jako je počet stavů a orientovaných vazeb modelu, a také zvolím vhodný typ parametrů. Už vím, že se bude jednat o MFCC koeficienty zejména jde o stanovení druhu rozdělení výstupní pravděpodobnosti. Druhý způsob se zakládá na metodě statistické indukce z množiny trénovacích dat, čímž poskytuje odhad hodnot parametrů HMM.

4.2.2 Skrytý Markovský model

HMM je model stochastického procesu, na nějž může být nahlíženo jako na pravděpodobnostní konečný automat, který v diskrétních časových okamžicích generuje náhodnou posloupnost pozorování $X = \{x_1, x_2, \ldots, x_T\}$.

Přechodová matice A je čtvercová o rozměrech N, N, kde N je počet stavů modelu, a každý prvek matice a_{ij} má význam podmíněné pravděpodobnosti přechodu modelu ze stavu s_i, v kterémkoliv čase t, do stavu s_j v čase $t + 1$.

$$a_{ij} = P(s(t + 1) = s_j|s(t) = s_i), \tag{28}$$

kde $s(t)$ je stav modelu v čase t. Pravděpodobnosti a_{ij} jsou konstantní v čase a musí pro všechny stavy $s_i, \forall i$ splňovat podmínku

$$\sum_{j=1}^{N} a_{ij} = 1. \tag{29}$$

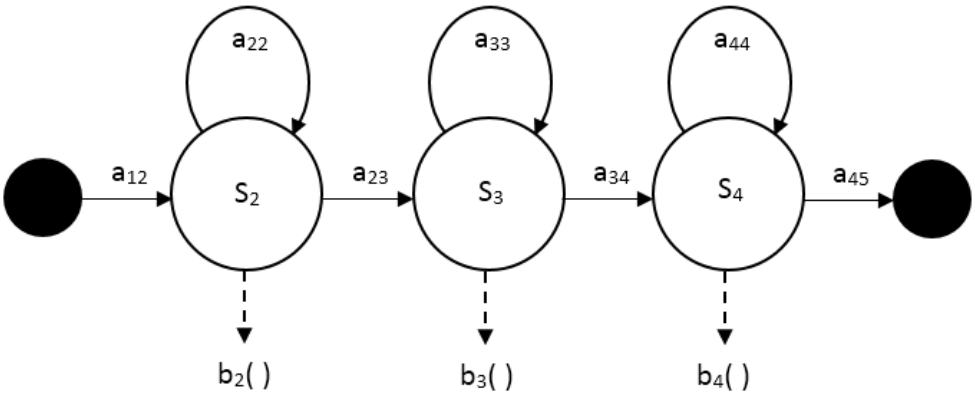
Stav s_j, do kterého model přejde, generuje příznakový vektor pro pozorování x_t, a to podle rozdělení výstupní pravděpodobnosti $b_j(x_t)$ příslušné k tomuto stavu [2].

Při trénování HMM modelováním mluvené řeči se využívají zejména tzv. „levo-pravé“ Markovské modely, které jsou zvláště vhodné pro modelování procesů, jejichž vývoj je spojen s postupujícím časem. Základní vlastností těchto modelů je to, že proces začíná příchodem prvního spektrálního vzoru z počátečního stavu modelu a se vzrůstajícím časem dochází k průchodům ze stavů s nižšími indexi do stavů s vyššími indexi nebo k setrvání ve stejném stavu. Průchod modelem je tedy z leva do prava. Proces končí příchodem posledního vektoru příznaků, přičemž model se v tomto okamžiku nachází v koncovém stavu. Pro „levo-pravé“ Markovské modely tak má matice A spicifický tvar

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & a_{22} & a_{23} & 0 & & 0 \\ 0 & 0 & a_{33} & a_{34} & & 0 \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & a_{N-1N-1} & a_{N-1N} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{30}$$

z něhož můžeme vidět, že krajní stavy slouží pouze ke spojování jendotlivých HMM, nazývají se neemitující a jsou v topologii modelu automaticky zahrnuty, z toho důvodu platí, že počet stavů $N > 2$, obvykle se ale volí z množiny lichých čísel tzn. $3, 5, 7$.

Na strukturu modelu je kladeno množství kritérií tak aby co nejvěrněji popisovala lidskou řeč. Vedle jednoduchosti, která posiluje robustnost odhadů (čím méně parametrů, tím robustnější může být jejich odhad) je dalším obvyklým požadavkem, aby každý stav reprezentoval určitou kvazistacionární část fonému. Toho, může být do určité míry dosaženou zvolením pětistavového modelu, znázorněném na obrázku 3. Kde vnitřní emitující stavy modelují náběhovou, střední a doběhovou část fonémů.



Obrázek 3: Ukázka pětistavového HMM

Funkce $b_j(x_t)$ popisuje rozdělení výstupní pravděpodobnosti modelu. Pokud je pozorování x_t vektorem spojitých náhodných proměnných, představují funkce $b_j(x_t)$ hustoty pravděpodobnosti jevu, že stav s_j v čase t generuje pozorování x_t neboli

$$b_j(x_t) = P(x_t|s(t) = s_j). \tag{31}$$

4.3 Volba metriky

Vyházeje z předchozích kapitol, volím pro můj návrh detektoru řeči za vhodnou metriku právě funkční hodnotu výstupních hustot pravděpodobností $b_j(x_t)$ (popsané v kapitole 4.2.2)skrytých Markovských modelů pro vektor pozorování x_t reprezentovaný MFCC koeficienty. „Rozdělení výstupní pravděpodobnosti musí být dostatečně specifické, aby od sebe oddělilo různé zvuky, a zároveň dostatečně robustní, aby zahrnulo značnou variabilitu řečového signálu. V současné době je nejvíce využíváno spojité rozdělení se směsí normálních (Gaussovských) funkcí hustot pravděpodobnosti. Spojité rozdělení se směsí normálních hustotních funkcí, je rozdělení, kde tvar výstupní hustoty pravděpodobnosti je tvořen váženým součtem jednotlivých normálních hustot pravděpodobností, z nichž každá je určena svým vektorem středních hodnot a svou kovarianční maticí“ [2].

Jednotlivé MFCC koeficienty jsou tak vnímány jako Gaussovské náhodné proměnné a díky vzájemné nekorelovanosti je možné je považovat za asymptoticky nezávislé. To mi umožňuje uvažovat kovarianční matici pouze jako diagonální, což mimo jiné významně snižuje výpočetní náročnost algoritmu.

Parametry rozdělení výstupní hustoty pravděpodobnosti jsou tak váhy jednotlivých složek neboli mixů, vektory středních hodnot a diagonální kovarianční matice.

Pro aktuální rámec ve zpracovávané promluvě tak vždy VAD algoritmus vypočte hodnotu metriky pro všechny emitující stavy všech HMM obsažených v použitém akustickém modelu. Vzniká tak matice mezivýsledků Met o rozměrech $dim(Met) = I \times J$, kde I má význam počtu všech HMM v akustickém modelu a J je počet emitujících stavů ve zvolené topologii Markovských modelů.

Výpočet funkční hodnoty multidimenzionální hustoty pravděpodobnosti Gaussovské směsi pro vektor pozorování $\vec{x_t}$ je definován následovně

$$Met_{ij}(\vec{x_t}) = b_{ij}(\vec{x_t}) = \sum_{m=1}^{M} \omega_{ijm} \cdot N(\vec{x_t}; \vec{\mu}_{ijm}, \Sigma_{ijm}), \tag{32}$$

kde i je index zpracovávaného HMM, j je index emitujícího stavu tohoto modelu, M je počet mixů v Gaussovské směsi, ω_{ijm} je váha m-tého mixu, kde platí

$$\sum_{m=1}^{M} \omega_{ijm} = 1, \tag{33}$$

a nakonec $N(\vec{x_t}|\vec{\mu}_{ijm}, \Sigma_{ijm})$ je multidimenzionální normální rozložení ve tvaru

$$N(\vec{x_t}|\vec{\mu}_{ijm}, \Sigma_{ijm}) = \frac{1}{(2\pi)^n|\Sigma_{ijm}|} e^{-\frac{1}{2}(\vec{x_t}-\vec{\mu}_{ijm})^T\Sigma_{ijm}^{-1}(\vec{x_t}-\vec{\mu}_{ijm})}, \tag{34}$$

kde $\vec{\mu}_{ijm}$ je vektor středních hodnot a Σ_{ijm} je kovarianční maticí mixu m stavu j a modelu i. Výpočet determinantu matice Σ_{ijm} je pro diagonální matici degradován na tvar

$$|\Sigma_{ijm}| = \prod_{k=1}^{K} \Sigma_{ijm}(k,k), \tag{35}$$

kde K je dimenze vektoru pozorování $dim(\vec{x_t})$. Podobně se zjednoduší i výpočet inverzní kovarianční matice

$$\Sigma_{ijm}^{-1}(k,k) = \frac{1}{\Sigma_{ijm}}(k,k). \tag{36}$$

Jelikož jsou hodnoty funkcí hustoty pravděpodobnosti pro jednotlivé vektory pozorování mnohem menší než jedna $b_{ij}(\vec{x_t}) << 1$, je vhodné s nimi dál pracovat v logaritmickém měřítku

$$lMet_{ij}(\vec{x_t}) = log(Met_{ij}(\vec{x_t})) = log(b_{ij}(\vec{x_t})). \tag{37}$$

4.4 Rozhodovací pravidlo

Rozhodovací funkce či klasifikační pravidlo je samotným jádrem detektoru řeči. V této práci budu používat několik, zde představených, variant a poté experimentálně ověřovat jejich funkci a vliv na kvalitu klasifikace řečového signálu ve vybraných akustických prostředích. Podoba rozhodovacího pravidla je do značné míry závislá na výběru metriky $lMet$, definované v předchozí kapitole, protože v podstatě jde o zpracování této matice do podoby, ze které je schopný rozhodnout o výskytu řeči. Může tak jít jak o jednoduchý výběr, tak o rozsáhlý rozhodovací strom.

- Max:
  Výchozí volbou je rozhodovací pravidlo založené, jak napovídá název, na jednoduchém výběru maximální hodnoty z matice metriky $lMet$ pro aktuálně zpracovávaný rámec. V tomto matici je na každé pozici uložena hodnota reprezentující hodnotu kritéria jednoho emitujícího stavu konkrétního Markovského modelu. Díky tomu jsem schopen určit s jakým HMM vybraná maximální hodnota koresponduje. Podle toho zda danný HMM reprezentuje jednotku řeči či šumu dochází následně ke klasifikaci rámce.

  $$Klas_{Max} = \begin{cases} 1 & \text{pro } arg\max_{i,j}\{lMet_{ij}\} \in S, \\ 0 & \text{pro ostatní,} \end{cases} \tag{38}$$

  kde i je index HMM v akustickém modelu, j index jeho emitujícího stavu a S je množina stavů náležících do HMM řeči v akustickém modelu.

- SumMax:
  Toto pravidlo modifikuje předchozí pravidlo Max. Podobným způsobem hledá maximum ale tentokrát ale ne přímo v matici metriky $lMet$. Jak již jsem zmínil u pravidla Max, každá hodnota v této matici odpovídá jednomu stavu HMM. Pravidlo $SumMax$ sečte hodnoty patřící stavům jednoho modelu a tímto způsobem pokřačuje i pro ostatní modely. Z těchto sum následně vybírá maximum. Dále už je postup totožný s pravidlem Max.

  $$Klas_{SumMax} = \begin{cases} 1 & \text{pro } arg\max_{i,j}\{\sum_{j=1}^{J} lMet_{ij}\} \in S, \\ 0 & \text{pro ostatní,} \end{cases} \tag{39}$$

- FirstK
  Předchozí pravidla fungovali pouze na principu výběru jedné hodnoty, buď přímo z matice metriky $lMet$ nebo jejím částečným zpracováním.

Toto pravidlo vybírá K nejvyšších hodnot z $lMet$ a následně vypočte četnost výskytu pro Markovské moduly představujících řeč a šum v této K-tici. Podle početní převahy té či oné skupiny je rozhodnuto o výsledné klasifikaci, v případě shody se pravidlo přiklání ke klasifikaci řeči.

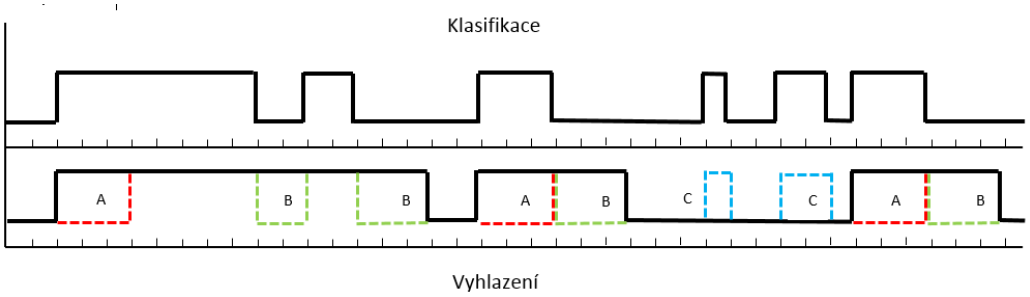
4.5 Vyhlazení výstupních dat

Dále je, pro zajištění vyšší přesnosti a plynulosti klasifikace, rozhodnutí klasifikátoru modifikováno skrze upravené schéma Hangover, popsané v kapitole 2.3.1. Používá se pro minimalizaci chybné detekce řeči ve slabých koncích promluv, kdy začátek vyslovovaného fónu je obvykle spojen (obzvláště u tzv. exploziv) se značným nárůstem energie v signálu, která ale rychle vyprchává a může tak docházet k chybám.

Dosahuje se toho použitím principu úmyslného zpoždění klasifikace při přechodech z úseků řeči do ticha jak je vidět na obrázku 4 na úsecích označených písmenem B. Kdy v první výskyt popisuje situaci, kdy v delším časovém úseku jsou rámce klasifikovány jako řeč a objeví se úsek o délce menší než parametr metody U, vyjadřující velikost zpoždění, následován opět alespoň jedním rámcem klasifikovaným jako řeč. V této situaci dochází ke zmíněnému vyhlazení rozhodnutí a úsek ticha je překlasifikován na řeč. Další výskyty popsané písmenem B představují situaci kdy dochází pouze ke zpoždění kvalifikace o U rámců v přechodu řeči na ticho i v případě, že nenavazuje opět úsek řeči.

Dalším efektem použití metody Hangover v této mnou rozšířené podobě je vyhlazení finální klasifikace i v opačných přechodech z ticha do řeči jako jsou úseky popsané písmeny A a C. Pro názornost opět popíši ukázkovou situaci, kdy se v případech delšího ticha vyskytne úsek řeči. Při procesu vyhlazení tak dochází k dočasnému zpoždění finální klasifikace a pokud je počet rámců v úseku řeči, stejně jako v C menší než je hodnota parametru metody D, mající taktéž význam zpoždění, je zmíněný úsek řeči překlasifikován na ticho. Pouze pokud je úsek delší zůstává původní klasifikace, jak je vidět v úsecích označených písmenem A, aby nedocházelo k nežádoucímu zpožďování klasifikaci na začátcích promluv.

Z popisu funkce lze vyvodit, že tato metoda je ve sporných situacích více nakloněna klasifikaci rámců jako řeči. Toto chování je ale žádoucí, protože zásadnější je pro tento druh úlohy VAD odhalit maximum řeči v promluvě i za cenu nesprávné klasifikace u krátkých úseků ticha, jako jsou například nádechy, mezislovní pauzy, mlasknutí atp., a eliminovat hlavně dlouhá ticha.



Obrázek 4: Ukázka vyhlazení výstupu klasifikátoru upravenou metoudou Hangover pro parametry $U, D = 3$

K náhledu je v příloze (I) příklad zdrojového kódu v programovacím jazyce Matlab, který přesně popisuje funkci mnou upravené metody Hangover, kde jsou vstupními parametry U a D, které odpovídají počtu rámců o kolik by se mělo opozdit rozhodnutí „neřeč $\rightarrow$ řeč“ nebo naopak.

4.6 Vyhodnocení kvality detekce řeči

V této kapitole bych chtěl popsat jakým způsobem přistupuji k vyhodnocení kvality mnou navrženého robustního detektoru řeči. Protože, pojem „robustní“ může být chápán z více hledisek, měl bych upřesnit definici tohoto pojmu pro úlohu detekce řeči. Tedy, VAD je robustní pokud poskytované rozhodnutí je blízké referenčnímu, představující ideální výstup detektoru, pro libovolné prostředí. Z toho vyplývá, že robustnost VAD může být odhadnuta aplikací detektoru řeči na nahrávku z prostředí s různými charakteristikami šumu v pozadí a porovnáním výsledků s referenční informací. Výkon VAD tak bude ohodnocen nejen obvyklou přesností SCR (Succes Rate) a aktivitou detekované řeči S_{klas} ale i robustností odvozenou od porovnání přesnosti pro různá prostředí. Čím jsou rozdíly v přesnosti menší tím robustnější je VAD.

Při výpočtu přesnosti budu detekovat dva druhy chyb s jistou mírou tolerance T vůči jejich výskytu udávanou v počtu rámců. Tolerance při výpočtu přesnosti je použita z důvodu, aby nedocházelo k přílišnému poklesu přesnosti klasifikace u nahrávek s častými přechody mezi řečí a šumem, v situacích kdy je rozhodnutí klasifikátoru na okolí T pouze posunuto, rozšířeno nebo naopak zúženo. To má za následek, že za špatnou klasifikaci budou pro vyhodnocení považovány pouze zásadní odchylky od reference.

Tolerované nepřesnosti mají minimální vliv z hlediska funkčnosti detektoru řeči, nicméně pokud by byly započteny zkreslovali by četnost výskytu vážnějších diferencí. Při volbě vhodné míry tolerance T se doporučuje brát ohled na požadovanou přesnost aplikace. Nicméně doporučuji vybírat hodnoty v řádu jednotek, maximálně desítek rámců, kde při velikosti rámce $10[ms]$ a volbě $T = 10$ jsou tolerovány nepřesnosti v okolí o poloměru $0.1[s]$.

Zaznamenávané chyby při vyhodnocení výkonu navrženého detektoru řeči:

- SAN (Speech As Noise):
  Klasifikace řečových úseku jako neřeč či šum, dle reference.
- NAS (Noise As Speech):
  Klasifikace neřečových úseku jako řeč, dle reference.

Rámec v čase k je vyhodnocen za správně klasifikovaný pokud v rámcích na okolí $\langle k - T, k + T\rangle$ došlo ke shodě referenční hodnoty s klasifikací pro rámec k. V podstatě tedy vyhledávám výstupní hodnotu klasifikátoru (údaj symbolizující buď řeč či neřeč) v referenčních datech v tolerovaném okolí od vyhodnocovaného rámce. Údaj SCR vyjadřující procentuální zastoupení

správně klasifikovaných rámců v promluvě je definován v rovnici níže, kde N_{scr} je počet správně klasifikovaných rámců a N_{fr} je počet všech rámců v promluvě.

Pokud na tomto okolí nedošlo ke shodě je rámec klasifikován jako chybně klasifikovaný a dále se určuje jakého typu tato chyba je. V případě, že hodnota klasifikátoru reprezentuje ticho a rámec je vyhodnocen za chybně klasifikovaný, jedná se o chybu typu SAN, jejíž minimalizace je pro tuto úlohu zásadní. Představuje totiž procentuální část nahrávky ve které se mluví a která není zpracovávána systémem pro automatické titulkování, z čehož plynou chyby v titulkách či úplně vynechané úseky.

V opačném případě je to logicky chyba typu NAS, která je sice méně závažná ale zajisté není zanedbatelná pro tuto úlohu. Toto vyhodnocení jsem navrhl taky aby součet SCR, NAS, SAN dal dohromady vždy 100%. Vzorce definující výpočet těchto údajů jsou

$$SCR = \frac{N_{scr}}{N_{fr}} \times 100, \tag{40}$$

$$SAN = \frac{N_{san}}{N_{fr}} \times 100, \tag{41}$$

$$NAS = \frac{N_{nas}}{N_{fr}} \times 100. \tag{42}$$

Algoritmus pro vyhodnocení úspěšnosti VAD algoritmu je k nahlédnutí v příloze (II).

Přestože metoda popsaná výše poskytuje užitečné objektivní informace týkající se výkonu VAD, pokud se má nasadit do reálného prostředí pro obecné použití nejčastěji se používá způsob vyhodnocení subjektivními testy, s cílem zabezpečit, že ořezávání bude vnímáno jako přijatelné. Tento typ testu požaduje přesný počet posluchačů k posouzení nahrávek obsahujících zpracované výsledky testovaného VAD. Posluchači musí oznámkovat následující kritéria.

- Kvalita.
- Obtížnost porozumění.
- Slyšitelnost oříznutí.

Tyto známky, získané poslechem několika řečových sekvencí, jsou dále použity pro výpočet průměrného výsledku pro každé výše jmenované kritérium. To poskytuje globální odhad chování testovaného VAD. Závěrem,

zatímco objektivní metody jsou velmi užitečné v počáteční fázi vyhodnocení kvality VAD, subjektivní metody hodnocení jsou více vypovídající. Protože subjektivní testy jsou dražší a vyžadující účast určitého počtu lidí po několik dní, nebudu jich v této práci využívat i přes to, že jsou jediné všeobecně používané pro standardizaci VAD algoritmů.

5 Implementace

V této kapitole popisuji konkrétní postup manipulace s daty včetně zvolených hodnot parametrů od stavu kdy mám pouze vstupní data, ve kterých chci řeč detekovat, až k získání výsledného rozhodnutí.

Vstupy do popisovaného detektoru řeči jsou textový soubor se seznamem nahrávek, parametrizované promluvy ve kterých se má detekce provést ve formátu `htk` a natrénovaný akustický model.

Pro parametrizaci audio nahrávek a trénování akustického modelu používám modul HTK (Hidden Markov Model Toolkit), jenž obsahuje rozsáhlou univerzální sadu nástrojů, ovladatelnou z příkazové řádky, zaměřenou na vytváření HMM a manipulaci s nimi. Podrobnější informace o tomto modulu je možné nalézt v dokumentu [11].

5.1 Parametrizace nahrávek

Pro získání parametrizovaných nahrávek, potřebných jak pro samotnou detekci řeči tak i pro původní natrénování akustického modelu na trénovacích datech slouží v modulu HTK program `HCopy`. Ten podle zadaných vstupních parametrů dokáže vyčíslit různé druhy posloupností koeficientů. Kromě mnou používaných MFCC jsou to například i PLP či LPC koeficienty.

```
HCopy -T 1 -C CF_param.mfc -S param.scp
```

`-T 1`	Hodnota představuje detailnost výpisů.
`-C CF_param.mfc`	Konfigurační soubor.
`-S param.scp`	Seznam promluv pro parametrizaci.

Konfigurační soubor `CF_param.mfc` obsahuje parametry, popisující strukturu a vlastnosti výsledných koeficientů. Soubor `param.scp` obsahuje seznam nahrávek kde každá řádka je rozdělena do dvou sloupců kde první definuje cestu ke zdrojové nahrávce a druhý cestu kam se mají parametrizované soubory uložit a pod jakým názvem.

Příklad použitého konfiguračního souboru s popisky parametrů uvádím k nahlídnutí níže. Zjednodušeně definuje, že vstupní soubory jsou ve formátu `WAV`, výstup bude vektor MFCC s delta a akceleračními koeficienty o dimenzi $dim(\vec{x}) = 3 \cdot 13 = 39$, za použití krátkodobé energie jako nultého koeficientu.

Velikost rámce stanovuje na $10[ms]$ a protože je vzorkovací frekvence $f_{vz} = 8kHz$ je počet filtrů v Melovské bance nastaven, podle tabulky(1), na 15.

```
# Coding parameters
SOURCEFORMAT = WAVE          %format zdrojoveho souboru
SOURCEKIND = WAVEFORM
TARGETKIND = MFCC_0_D_A      %0 - nulty koeficient energie signalu;
                             %D - delta koef; A - akcelerační koef;
TARGETRATE = 100000.0        %velikost ramce 10ms
SAVECOMPRESSED = F
WINDOWSIZE = 320000.0        %velikost okenka 32ms
USEHAMMING = T               %pouziti hamingova okénka
USEPOWER = T                 %pouziti energie
PREEMCOEF = 0.97               %koeficient preemfaze
NUMCHANS = 15                  %pocet filtru v Melovske bance
CEPLIFTER = 0                  %powerlifting koeficientu
NUMCEPS = 12                   %pocet koeficientu, + 1 pokud energie
```

5.2 Proces trénování akustického modelu

Posledním vstupem, který pro implementaci VAD algoritmu potřebuji je natrénovaný akustický model. Vzhledem k aplikovanému přístupu, popsaném v kapitole 4 , se tento akustický model vytvoří sloučením dvou samostatně natrénovných submodelů.

Každý z nich, bude obsahovat množinu HMM popisujících spektrální charakteristiky prvků jejichž seznam je pro každý akustický model uveden v souboru `monophones`. U prvního z nich je seznamem výčet fonémů českého jazyka. U druhého je to pak seznam všech uvažovaných šumů v různých akustických prostředích, viz 3.2.

Před samotným započetím procesu trénování pomocí modulu HTK potřebujeme následující výčet prerekvizit.

`train.scp`	Seznam trénovacích promluv.
`words.mlf`	Transkripce všech trénovacích promluv na úrovni slov.
`dict.txt`	Slovník výslovnosti.
`monophones`	Seznam symbolů fonetické abecedy.

Dále tak budu popisovat tvorbu a trénování obou submodelů zvlášť. Nejdříve si však nadefinuji pro oba společné parametry topologie HMM.

5.2.1 Zvolená topologie HMM

Vycházeje z informací z kapitoly 4.2.2, zvolil jsem použití pěti stavového monofónního tzv. „levo-pravého“ Markovského modelu, k vidění na obrázku (3). Jehož výstupní hustota pravděpodobnosti b_j, popsaná rovnicemi (33),(35), je pro každý emitující stav j multidimenzionální směsí normálních (Gaussovských) hustot pravděpodobnosti, kde počet dimenzí odpovídá velikosti vektoru MFCC koeficientů $dim(\vec{x}) = 39$ a počet mixů ve směši byl experimentálně optimalizován na $M = 13$ s ohledem na maximální přesnost následného rozpoznávání řeči, kde každý mix je jednoznačně určen váhou složky, vektorem středních hodnot a diagonální kovarianční maticí.

Výchozí hodnoty prvků matice přechodů A byli před trénováním modelů stanoveny následovně.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0.6 & 0.4 & 0 & 0 \\ 0 & 0 & 0.6 & 0.4 & 0 \\ 0 & 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{43}$$

Prototyp takto nadefinovaného ale pouze jednosložkového HMM je uložený v souboru `proto` a je dále zpracováván programem `HCompV`, který nastaví hodnoty vektoru středních hodnot a diagonální kovarianční matice. Využívá k tomu celkové průměrné hodnoty ze všech dat z trénovací množiny. Toto přednastavení slouží k urychlení a trénování modelů HMM.

```
HCompV -C CF.mfc -f 0.01 -m -S train.scp -M hmm0 proto
```

`-C CF.mfc`	Konfigurační soubor.
`-f 0.01`	Vytvoří makro vFloor1.
`-m`	Signalizuje vypočtení nejen kovariance ale i střední hodnoty.
`-S train.scp`	Seznam trénovacích promluv.
`-M hmm0`	Adresář pro uložení výsledného souboru.
`proto`	Definovaný prototyp.

5.2.2 Trénování modelu řeči

Protože, navržený VAD zaměřen na následné použití v aplikacích rozpoznávání řeči můžeme použít pro obě úlohy stejný akustický model. To je výhodné zvláště z důvodu snížení výpočetních nároků. Postup pro trénování akustického modelu řeči k tomuto účelu se tak skládá z následujících kroků.

- Vytváření souborů s přepisem na úrovni fonémů

  Abych mohl natrénovat akustický model na úrovni fónů jak potřebuji, musím nejdříve převést přepis nahrávek na úrovni slov `words.mlf` na přepis na úrovni fónů `phones.mlf`. Ten vytvořím pomocí programu `HLEd`, obsaženém v balíku HTK. Tento program obecně slouží k manipulaci se soubory `MLF`.

  ```
  HLEd -l * -d dict.txt -i phones.mlf mkphones.led words.mlf
  ```

`-l *`	Znak * se přidá do jmen souborů v MLF souboru místo skutečné cesty
`-d dict.txt`	Načte slovník výslovnosti.
`-i phones.mlf`	Výstupní soubor s transkripcí na úrovni fónů.
`mkphones.led`	Soubor s příkazy, které se mají provést.
`words.mlf`	Soubor s transkripcí promluv na úrovni slov.

- Vytvoření akustického modelu

  Pro každý symbol v souboru `monophones` je nutné vytvořit kopii do adresáře `hmm0` do tzv. MMF (Master Macro File), který tak bude obsahovat definice všech modelů a bude sloužit pro reprezentaci akustického modelu. Tento úkol řeším příkazem `MakeMMF`.

  ```
  MakeMMF proto monophones vFloors models
  ```

`proto`	Vstupní prototyp modelu.
`monophones`	Seznam monofónů.
`vFloors`	Obsahuje dolní mez kovariance.
`models`	Výstupní MMF soubor s modely monofónů.

- Trénování akustického modelu

  Nyní už mohu přistoupit k samotnému trénování modelu. Tuto úlohu má na starost program `HERest`, který je založen na „forward-backward“ algoritmu (viz.[11]). Program vyhledává v `hmm0/MODELS` modely podle názvů, uvedených v `monophones`. Při nalezení se model aktivuje, reestimuje použitím trénovacích dat v `train.scp` a opět se uloží tentokrát do složky `hmm1`.

```
HERest -T 1 -C CF.mfc -I phones.mlf -t 250.0 150.0 1000.0
       -S train.scp -H hmm0/MODELS -M hmm1 monophones
```

`-C CF.mfc`	Konfigurační soubor.
`-I phones.mlf`	Soubor s monofónní transkripcí trénovacích promluv.
`-t 250.0 150.0 1000.0`	Nastavuje práh prořezávání ve forward-backward algoritmu.
`-S train.scp`	Seznam trénovacích promluv.
`-H hmm0/MODELS`	Vstupní soubor MMF.
`-M hmm1`	Adresář pro uložení výsledného souboru.
`monophones`	Seznam monofónů čili seznam jmen modelů.

  Pro kvalitní natrénování modulů HMM je vhodné trénování několikrát opakovat, přičemž vstupní parametry se liší pouze ve vstupních a výstupních adresářích.

- Přerovnání trénovacích dat

  Smysl přerovnání trénovacích dat zjednodušeně spočívá v překladu transkripce na úrovni slov na transkripci na úrovni monofónů. Pokud ve slovníku existuje více možností překladu daného slova, vybere se varianta, která nejvíce odpovídá trénovacím datům a natrénovaným modelům. Toho se dosahuje využitím programu `HVite`, založeném na Viterbiho algoritmu.

```
HVite -T 1 -l * -y lab -o SWT -b _SIL_ -C CF.mfc -m -a -H
      hmm4/MODELS -i aligned.mlf -t 250.0 -I words.mlf -S
      train.scp dict.txt monophones
```

`-l *`	Znak * se přidá do jmen souborů v MLF souboru místo skutečné cesty
`-y lab`	Koncovka jmen souborů v MLF souboru.
`-o SWT`	Formát výstupního MLF souboru.
`-b _SIL_`	Vkládá slovo *_SIL_* na začátek a konec každé promluvy.
`-C CF.mfc`	Konfigurační soubor.
`-m`	Nastavuje výstup na úrovni fonémů.
`-a`	Zajišťuje provedení přerovnání trénovacích dat.
`-H hmm4/MODELS`	Vstupní soubor MMF.
`-i aligned.mlf`	Výstupní přerovnaný soubor MLF.
`-t 250.0`	Nastavuje práh prořezávání pro „beam search“.
`-I words.mlf`	Vstupní soubor MMF.
`-S train.scp`	Seznam trénovacích promluv.
`dict.txt`	Rozpoznávací slovník.
`monophones`	Seznam monofónů čili seznam jmen modelů.

  Nahrávky, které se nepodaří zarovnat se do `aligned.mlf` neukládají a musí se odstranit i z `train.scp` programem `CreateAligned`.

```
CreateAligned.exe aligned.mlf train.scp aligned.scp ne.scp
```

`aligned.mlf`	Vstupní soubor s přerovnanou transkripcí. trénovacích promluv
`train.scp`	Vstupní seznam trénovacích promluv. promluv.
`aligned.scp`	Výstupní soubor se seznamem dobře přerovnaných promluv.
`ne.scp`	Výstupní seznam nepřerovnaných promluv.

Následně se modely HMM opět několikrát přetrénují pouze s tím rozdílem, že soubor s přepisem promluv na úrovni fónů `phones.mlf` je nahrazen zarovnaným přepisem `aligned.mlf`.

- Vytvoření Gaussovské směsi

  V tomto stavu máme natrénovaný jednosložkový akustický model. Posledním krokem je přidání a natrénování dalších složek programem `HHed`.

```
HHEd -T 1 -A -C CF.mfc -H hmm8_1\models -M hmm8_2
     add_next.hed ..\monophones
```

`-C CF.mfc`	Konfigurační soubor.
`-H hmm8_1/MODELS`	Vstupní soubor MMF.
`-M hmm8_2`	Adresář pro uložení výsledného souboru.
`add_next.hed`	Seznam příkazů pro vykonání.
`monophones`	Seznam monofónů čili seznam jmen modelů.

Pro každý znak v souboru `monophones` vyhledá ve vstupním modelu odpovídající HMM pro něj provede příkazi uvedené v `add_next.hed` a uloží ho do nového umístění. V mém případě se jedná o jediný příkaz, říkající přidej jednu složku ke stavům 2 až 4.

```
MU +1 {*.state[2-4].mix}
```

Následně se opět musí modely přetrénovat programem `HERest`. Toto se opakuje dokud není dosaženo optimálního počtu složek vzhledem k výsledkům rozpoznávání. Pro zjednodušení jsem si vytvořil skripty v jazyce `batch`, který celý proces automatizují.

5.2.3 Trénování modelu šumu

Proces trénování akustického modelu šumu provází většina kroků jako je tomu u modelu řeči. Proto zde popisuji pouze rozdíly mezi nimi.

Zásadní rozdíl tkví v nutnosti rozdílného předzpracování dat pro trénování.

K nahrávkám šumu přistupuji tak, že si je idealizuji a pracuji s nimi jako by každá obsahovala jediný zvuk. Každou nahrávku poté označím symbolem v `monophones` podléhajím vlastní konvenci značení. Toto označení je složeno z několika částí. První je informace, že se bude jednat o šum „_“, následuje zkratka kategorie prostředí do které zvuk patří (T,N,R) a poslední částí je pořadové číslo nahrávky v této kategorii.

Přepis na úrovni slov `words.mlf` tak pro každou nahrávku šumu obsahuje pouze jeden znak. Z toho vyplývá zbytečnost provádění kroku pro vytvoření souboru s přepisem nahrávek na úrovni fónů `phones.mlf`, protože by byl stejný. Dále odpadá i potřeba přerovnávání dat v promluvách. Proces trénování se tak zjednoduší do následujícího výčtu kroků.

- Vytvoření akustického modelu
- Trénování akustického modelu
  Pro případ, že v `phones.mlf` je pro nahrávky pouze jeden symbol je nutné přidat při spuštění programu další vstupní parametr `-m1` .

  ```
  HERest -m 1 -T 1 -C CF.mfc -I phones.mlf -t 250.0 150.0
       1000.0 -S train.scp -H hmm0/MODELS -M hmm1 monophones
  ```

- Vytvoření Gaussovského mixu

5.2.4 Sloučení akustických modelů

Výsledný akustický model sloužící jako jeden ze vstupů do navrženého VAD algoritmu se získá sloučením obsahu natrénovaného akustického modelu `models` pro řeč i šum.

5.3 Implementace VAD v Javě

Algoritmus detekce řeči implementuji skrze objektově orientovaný programovací jazyk Java ve vývojovém prostředí Eclipse.

Vlastní algoritmus VAD načítá natrénovaný akustický model spolu se seznamem již parametrizovaných nahrávek, který obsahuje relativní cestu k `htk` souborům. Podle vypočtené metriky a zvoleného rozhodovacího pravidla následně indikuje přítomnost řeči či nikoliv. Soubory s výsledky ukládá do vytvořených adresářů podle názvu aktuálně zpracovávané promluvy a zvolených parametrů.

```
VAD_algorithm models train.scp FirstK U 10 D 5 K 10 aligned.mlf
```

- popis vstupů:

`model`	Natrénovaný akustický model.
`train.scp`	Seznam s parametrizovaných nahrávek pro detekci řeči.
`FirstK`	Vybrané rozhodovací pravidlo.
`U10`	Parametr metody Hangover.
`D5`	Parametr metody Hangover.
`K10`	Parametr pravidla FirstK.
`aligned.mlf`	Soubor s referenční informací pro vyhodnocení.

- popis výstupů:

`res`	Adresář s výsledky detekce řeči ve formátu mlf.
`ref`	Adresář s referencí pro vykreslení v Matlabu.
`SCR`	Adresář s přehled ukazatelů kvality VAD pro danou nahrávku.
`mat`	Adresář s výsledky detekce řeči pro vykreslení v Matlabu.

6 Experimenty

V této kapitole popisuji experimenty prováděné na navrženém algoritmu detekce řeči nad testovací sadou popsanou v 3.3. Tyto experimenty jsou zaměřeny na ověření kvality VAD vůči referenční informaci a vyhodnoceny podle postupu popsaného v kapitole 4.6 s mírou tolerance pro vyhodnocení výsledků $T = 10$.

Smyslem těchto experimentů je snaha pokusit se objevit konfiguraci poskytující co nejkvalitnější výsledky. Především v nich budu zjišťovat a ověřovat vliv volby rozhodovacího pravidla klasifikátoru a jeho případných parametrů, včetně parametrů U a D vyhlazovací metody Hangover, na proces detekce řeči vzhledem k spektrálním charakteristikám okolního prostředí.

6.1 Reference pro vyhodnocení

Abych vůbec mohl posoudit kvalitu navrženého VAD potřebuji určitou referenční informaci, kterou mohu považovat za ideální výsledek a s touto informací porovnat data z výstupu VAD.

Pro úlohu detekce řeči se využívá několika možností jako tuto referenci získat, například ručním zarovnáním nebo použitím standardizovaného VAD. Ruční zarovnání je, pokud má být přesné, zdlouhavé a namáhavé. Pro standardizované VAD nejsou tak přesné abych je mohl považovat za ideální ale jsou jednoduché na použití. Vybral jsem ale způsob, který by měl spojovat obě výhody dohromady. Měl by být relativně rychlý a dosahovat vysoké přesnosti. Můj způsob spočívá v použití postupu popsaného při trénování akustického modelu 5.2, tentokrát ale s testovací sadou nahrávek. Nevýhodou tohoto postupu je nutnost vlastnit, kromě samotných nahrávek, minimálně přepis nahrávek na úrovni slov `words.mlf`. Z důvodu tohoto omezení, nebylo jednoduché vybrat vhodnou reprezentativní množinu nahrávek jako testovací sadu. Vysoká přesnost by ale měla být zajištěna právě díky této aditivní informaci kdy dopředu znám co bylo řečeno a je tak možné najít nejpravděpodobnější rozložení posloupnosti fónů v danné promluvě. Přerovnání promluv a získání referenční hodnoty probíhá pro každé akustické prostředí zvlášť.

Výsledná reference poté nabývá následující podoby a je uložena do souboru `aligned.mlf`.

```
#!MLF!#
"*/01_office.lab"
0 1200000 _sil_
1200000 44600002 _ns_
44600002 67500001 _sil_
67500000 69400000 z
69400000 70000000 U
70000000 71200000 s
71200000 71800000 t
.
```

První řádek je hlavička dokumentu značící, že jde o *MLF* soubor. Na druhé řádce je název zpracovávané promluvy s koncovkou *lab* uvozující zarovnání promluvy. Následují řádky rozdělitelné do třech sloupců. První dva označují začátek a konec časového intervalu na který je zarovnán symbol označující fón ve třetím sloupci. Jako základní jednotka času se uvažuje $10^{-7}[s]$, proto tak vysoká čísla. Aby byl referenční soubor načten správně musí všechny neřečové symboly začínat znakem „_“.

6.2 Experiment 1: Rozhodovací pravidlo Max

První experiment bude zaměřen na nastavení parametrů metody Hangover, viz kapitola 4.5 za použití výchozího klasifikačního pravidla maximální pravděpodobnosti Max. Hangover závisí na hodnotách parametrů U, D, které mají význam zpoždění rozhodnutí klasifikátoru v odlišných klasifikačních přechodech. Přechod z řečového úseku do neřečového se tak zpozdí o U rámců pokud se do té doby na vstupu metody Hangover neobjeví opět rámec klasifikovaný jako řeč. V tom případě by se čítač, kterým je toto zpoždění realizováno resetoval na počáteční hodnotu U. Podobně je tomu u parametru D pro opačné přechody z neřečových úseků promluvy do řečových. Maximální možné reálné zpoždění omezím apriori na $0,25[s]$. Z toho vyplývá, při délce rámce $10[ms]$, že interval hodnot, kterých tak parametry U a D mohou nabývat, je omezen na $\langle 0, 20\rangle$.

Experiment samotný je rozdělen do třech částí. Protože podrobné prohledávání i takto omezeného stavového okolí pro všechny kombinace akustických prostředí a parametrů U a D by bylo velice časově náročné ($3 \cdot 21^2 = 1323$ opakování), jsem nucen omezit výběr na výčet vybraných hodnot uvedených v množině $M = \{0, 2, 6, 10, 14, 18, 22\}$. Z důvodu výpočetní náročnosti jsem dále přistoupil k oddělené volbě parametrů. Budu tak nejdříve hledat suboptimální hodnotu parametru D pro fixní parametr $U = 10$, nastavený na medián množiny M.

6.2.1 Odhad parametru D

Pro každý prvek množiny M spustím algoritmus detekce řeči pro první (nastavovací) nahrávku z každé uvažované kategorie akustických prostředí. Následně vzájemně porovnám výkonnost jednotlivých nastavení podle procentuální úspěšnosti klasifikace SCR a velikosti relativní chyby SAN (Spech As Noise) a stanovím optimální hodnotu parametru D pro jednotlivé kategorie prostředí. Výběr zobrazení chyby SAN má opodstatnění ve vyšší důležitosti této chyby oproti NAS. Pro aplikace zpracování řeči, je vždy větší problém pokud detektor řeči příliš „ořízne“ řečové úseky zpracovávané promluvy, než když označí za řeč úsek ticha i když ani tato chyba není zanedbatelná. Algoritmus výpočtu ukazatelů kvality SCR, NAS, SAN, viz kapitola 4.6 a příloha (II).

	T		N		R	
D	SCR[%]	SAN[%]	SCR[%]	SAN[%]	SCR[%]	SAN[%]
0	96.71	2.55	90.61	1.72	85.98	13.38
2	94.48	5.01	91.00	3.49	76.44	23.22
6	89.64	10.23	85.95	11.29	61.57	38.29
10	87.59	12.40	79.86	18.28	56.96	42.98
14	86.80	16.97	75.58	23.19	56.01	43.98
18	86.61	17.21	71.08	28.05	55.08	44.91
22	86.42	17.46	68.08	31.33	54.93	45.06

Tabulka 3: Tabulka úspěšnosti detekce řeči a výskytu chyby typu SAN pro různé hodnoty D v závislosti na akustickém prostředí, pro fixní parametr $U = 10$ a za použití rozhodovacího pravidla Max.

V tabulce (3) je zřetelně vidět trend výrazně se zvyšující procentuální úspěšnosti detekování řeči i klesající chyby NAS pro všechna prostředí se snižující se hodnotou parametru D až k nule. Naopak pro hodnoty $D > 2$ jsou výsledky VAD nekvalitní natolik, že by nebylo možné je při tomto nastavení v praxi reálně použít. Tento jev se dá vysvětlit pozvolným přechodem z úseku ticha do úseku řeči na výstupu klasifikátoru a následného promíchání rámců klasifikovaných jako řeč a šum.

Pro tiché prostředí T pozoruji nejvyšší dosaženou úspěšnost ze všech akustických prostředí. To není nic překvapivého, protože v těchto prostředích jsou nejvíce patrné rozdíly mezi úseky řeči a šumu. U rušných prostředí R je situace opačná, protože zde je SNR naopak blízké nule. Pro obě tyto kategorie prostředí volím hodnotu $D_T = 0$, $D_R = 0$. Situace v běžných prostředí N je složitější. Četnost chyby SAN klesá se snižujícím se D opět až k nule ale

úspěšnost dosáhla vrcholu už krok dříve. V tomto případě dávám přednost důležitosti celkové úspěšnosti na úkor vyšší chyby SAN a volím parametr $D_N = 2$.

6.2.2 Odhad parametru U

Druhá část spočívá v opakování předchozího postupu tentokrát pro parametr U, kde parametr D je fixován na vybrané hodnotě v závislosti na předchozím kroku, pro každé prostředí zvlášť.

	T		N		R	
U	SCR[%]	SAN[%]	SCR[%]	SAN[%]	SCR[%]	SAN[%]
0	95.45	4.51	82.78	4.66	81.41	18.56
2	95.88	4.01	85.08	3.77	82.69	17.24
6	96.48	3.18	87.77	2.49	84.68	15.11
10	96.71	2.55	90.61	1.72	85.98	13.38
14	96.60	2.05	93.40	1.16	86.92	11.85
18	96.17	1.73	95.06	0.90	87.45	10.53
22	95.69	1.48	94.94	0.74	87.52	9.46

Tabulka 4: Tabulka úspěšnosti detekce řeči a výskytu chyby typu SAN pro různé hodnoty U v závislosti na akustickém prostředí za použití rozhodovacího pravidla Max.

Opět mohu v tabulce (4) ve většině případů pozorovat zřetelný vývoj hodnot s měnícím se U. Zatímco hodnoty chyb typu SAN se vzrůstající hodnotou parametru U klesají pro všechny tři kategorie u úspěšnosti SCR situace opačná, s výjimkou pro tiché prostředí T. Pro kategorii tichých akustických prostředí volím parametr $U = 14$, přičemž se snažím vyvážit protichůdné požadavky obou sledovaných kritérií. Podobně se snažím vybalancovat tato kritéria pro kategorii N volbou $U = 18$. A nakonec, pro rušná akustického prostředí volím $U = 22$.

6.2.3 Ověření odhadnutých hodnot parametrů

Nakonec vybrané kombinace hodnot parametrů U, D otestuji, postupně pro všechny definovaná akustická prostředí na druhé (testovací) nahrávce.

	U	D	SCR[%]	SAN[%]
T	14	0	95.25	2.74
N	18	2	91.96	1.60
R	22	0	86.75	11.26

Tabulka 5: Tabulka úspěšnosti detekce řeči a výskytu chyby typu SAN pro ověřovací data akutických prostředí a pro nalezené hodnoty parametrů metody Hangover U a D za použití rozhodovacího pravidla Max.

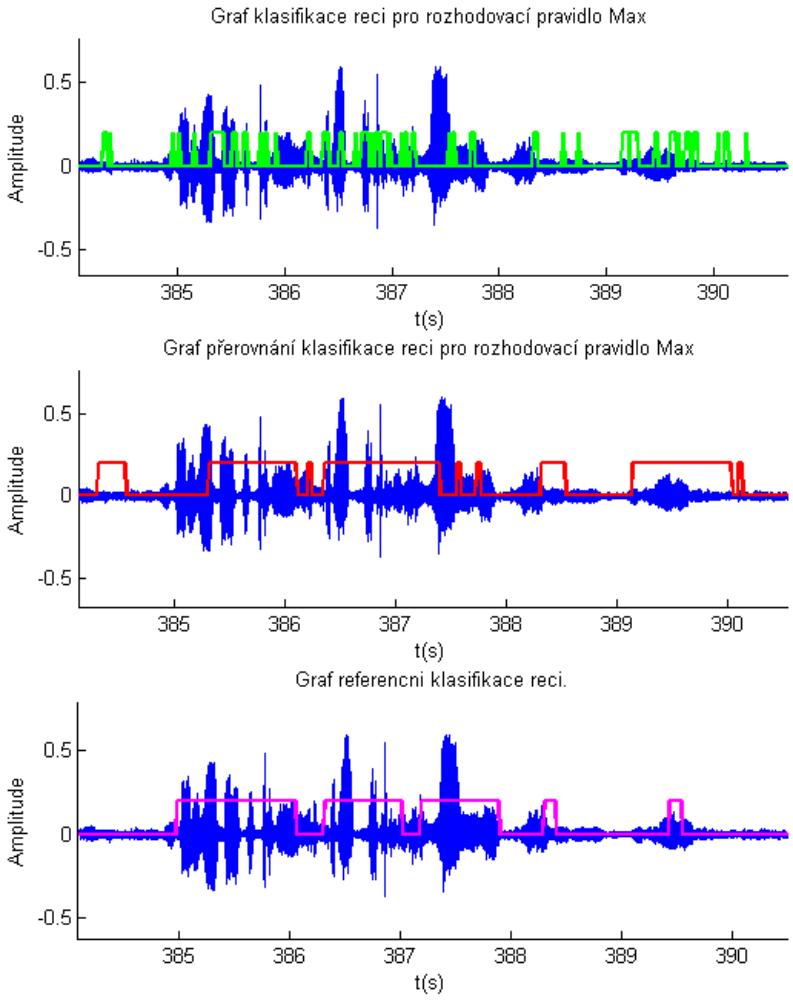
Na výsledcích uvedených v tabulce (5) je patrné, že pozorovaná kritéria výkonnosti VAD se pro testovací nahrávky oproti nastavovacím v tabulce (4) kvalitativně pohoršili při stanovených parametrech U a D. Rozdíly jsou ale stále nejsou významné. To je opět logicky zdůvodnitelné, protože i přes to, že nahrávky patří do stejné kategorie mohou pro ně výsledky mírně kolísat. Tento fakt, mi umožňuje případné použití takto zvolených parametrů na další nahrávky z těchto kategorií.

6.2.4 Vyhodnocení experimentu:

Pro metodu Hangover se mi podařilo určit vhodné hodnoty parametrů pro všechna akustická prostředí a následně ověřit stabilitu těchto výsledků na testovací sadě, jak je vidět v tabulkách (4) a (5). Hodnoty U a D byli stanoveny z výsledků v tabulkách na základě snahy o vyvážení maximalizace úspěšnosti detekce řeči SCR a minimalizace chyby typu SAN. Přesto mohou být zvolené hodnoty parametrů zarážející, protože se ukázala přímá úměrnost mezi velikostí parametru D a chybou SAN. To vedlo v případě běžného prostředí N k volbě $D = 2$, u tichého prostředí T a rušného prostředí R dokonce k úplnému potlačení zpoždění klasifikace pro přechody z neřečových úseků do řečových, $D = 0$. Dle údajů v tabulkách o úspěšnoti detekce řeči, poskytuje klasifikátor s rozhodovacím pravidlem Max uspokojivě kvalitní a přesné informace o výskytu řeči pro tichá akustická prostředí T a do značné míry i pro prostředí N. Výkon navrženého VAD algoritmu pro poslední nejdynamičtěji se měnící kategorii prostředí R, obsahující nahrávky s nejvýraznějším zastoupením šumu ale není přesvědčivý. Všeobecně je, ale zpracování nahrávek z takovéhoto druhu prostředí pro většinu dnes používaných VAD algoritmů problém.

6.2.5 Ukázka průběhu klasikace

Grafická ukázka detekování řeči pravidlem Max pro nalezené hodnoty parametrů $U = 18$ a $D = 2$ metody Hangover pro testovací nahrávku akustického prostředí N.



Obrázek 5: Ukázka VAD, pro pravidlo Max na akustickém prostředí N

6.3 Experiment 2: Rozhodovací pravidlo $SumMax$

Motivací k dalším pokusům je stále nepříliš vysoká kvalita detekce řeči pro rušná akustická prostředí R, spolu s paralelní snahou o nalezení možné kvalitnější metody detekování řečových úseků pro zbylá prostředí T, N.

V druhém experimentu se pokouším o zvýšení výkonu navrženého algoritmu VAD, změnou rozhodovacího pravidla, z Max na $SumMax$, jejichž funkce je upřesněna v kapitole 4.4. Budu tak moci zkoumat vliv volby rozhodovacího pravidla na výkon a robustnost navrženého algoritmu, pro různá akustická prostředí, s cílem zvýšit výkonnost VAD pro prostředí R při zachování vysoké kvalitativní úrovně detekce řeči pro prostředí T a N.

Zjednodušeně, klasifikační pravidlo $SumMax$ vypočítává pro každý model HMM obsažený v akustickém modelu hodnotu kritéria, které odpovídá sumě z matice metriky $lMet$ přes všechny emitující stavy konkrétního HMM. Následně z množiny HMM vybere ten s maximální hodnotou získaného kritéria a rozhoduje o výskytu řeči na základě toho zda vybraný model reprezentuje jednotku řeči či šumu. Oproti pravidlu Max tak bude zpracovávaný rámec přirovnávat k jednotlivým skrytým Markovovým modelům a ne jeho stavům.

Parametry metody pro vyhlazení dat na výstupu klasifikátoru Hangover nastavím podle výsledků z předešlého experimentu na hodnoty uvedené v tabulce (5). Před porovnáním výsledků je nutné nejdříve, stejně jako v předchozím experimentu, zjistit optimální hodnoty parametrů pro metodu Hangover, protože změna rozhodovacího pravidla mohla mít vliv na funkci této metody. Postup pro hledání suboptimálních hodnoty parametrů U a D, kde U má význam zpoždění detekce ticha na konci promluv a podobně jako D řeči na jejich začátcích, je shodný jako v přecházejícím experimentu. Z toho důvodu dále uvedu pouze tabulky s výsledky a jejich zhodnocení.

6.3.1 Odhad parametru D

	T		N		R	
D	SCR[%]	SAN[%]	SCR[%]	SAN[%]	SCR[%]	SAN[%]
0	93.61	6.08	89.22	1.18	88.88	7.81
2	89.76	10.14	89.31	3.03	82.27	15.46
6	87.04	12.95	83.32	12.15	66.25	32.74
10	86.46	13.53	76.52	20.12	60.05	39.57
14	86.37	13.62	72.68	25.31	57.67	42.14
18	86.37	13.62	68.62	29.91	56.40	43.52
22	86.37	13.62	66.41	32.88	55.84	44.10

Tabulka 6: Tabulka úspěšnosti detekce řeči a výskytu chyby typu SAN pro různé hodnoty D v závislosti na akustickém prostředí, pro fixní parametr $U = 10$ a za použití rozhodovacího pravidla $SumMax$.

V tabulce (6) je vidět, při hledání optimální hodnoty parametru D za fixní hodnoty parametru $U = 10$, ještě více než u pravidla Max, závislost mezi klesajícím D, roustucí úspěšností SCR a klesající chybou SAN. Pro kategorie akustického prostředí T a R je závěr jasný, protože pro obě se nejlepších výsledků dosahuje pro $D = 0$. K diskuzi by mohlo dojít u bežných akustických prostředí. Nicméně, pokud chceme vhodně vyvážit hodnoty kritických parametrů SCR a SAN, docházím k rozhodnutí zvolit i pro tuto kategorii hodnotu parametr $D = 0$. Toto rozhodnutí zdůvodňuji minimálním nárůstem přesnosti rozpoznávání SCR se značným nárůstem chyby SAN pro $D = 2$ oproti výsledkům při $D = 0$.

6.3.2 Odhad parametru U

	T		N		R	
U	SCR[%]	SAN[%]	SCR[%]	SAN[%]	SCR[%]	SAN[%]
0	92.02	7.91	95.27	4.17	86.91	13.03
2	92.42	7.52	94.93	3.21	88.08	11.58
6	93.17	6.73	92.44	1.86	88.95	9.44
10	93.61	6.08	89.22	1.18	88.88	7.81
14	93.90	5.35	86.51	0.74	88.65	6.57
18	94.10	4.78	84.14	0.50	88.36	5.52
22	94.24	4.21	81.71	0.34	87.68	4.77

Tabulka 7: Tabulka úspěšnosti detekce řeči a výskytu chyby typu SAN pro různé hodnoty U v závislosti na akustickém prostředí, za použití rozhodovacího pravidla $SumMax$.

V tabulce (7) jsou zřetelně vidět odlišné trendy vývoje kritérií vzhledem k měnícímu se U, výjimkou je pouze úspěšnost klasifikace SCR pro rušná prostředí. Přesto výběr optimálních hodnot pro paramer U není všude jednoznačný. Nejjasnější je situace u tichých prostředí T, kde dosahujeme shodně kvalitativně lepších výsledků se zvyšujícím se U a proto pro něj volím maximální hodnotu $U = 22$. Ne tak jasné je rozhodnutí pro rušná prostředí R, kvůli již zmíněnému vývoji hodnot úspěšnosti SCR. Přinucen ke kompromisu, zvolil jsem výslednou hodnotu $U = 18$, tak aby došlo k co nejvýraznějšímu snížení chyby typu SAN při minimálním snížení celkové přesnoti klasifikace. Nejsložitější volba však nastává při volbě v kategorii normálních akustických prostředí kde obě posuzovaná kritéria SCR i NAS dosahují kvalitativně velice dobrých hodnot ovšem pro opačné konce intervalu, ze kterých bylo voleno U pro prováděný experiment. Při vyvažování kritérií jsem došel k závěru, že optimální hodnota parametru $U = 6$.

6.3.3 Ověření odhadnutých hodnot parametrů

Stejně jako v prvním experimentu nalezené kombinace parametrů U a D otestuji pro všechny kategorie na ověřovacích nahrávkách. Činím tak abych se ujistil, že úroveň kvality dosažených výsledků je v určitých mezích stálá a není náchylná k výrazným změnám při aplikaci na jinou klasifikovanou nahrávku.

	U	D	$SCR[\%]$	$SAN[\%]$
T	22	0	91.75	6.03
N	6	0	92.03	3.04
R	18	0	90.36	6.80

Tabulka 8: Tabulka úspěšnosti detekce řeči a výskytu chyby typu SAN pro ověřovací data akutických prostředí a pro nalezené hodnoty parametrů metody Hangover U a D za použití rozhodovacího pravidla $SumMax$.

Výsledky ověřovacích nahrávek v tabulce (8) pro nalezené hodnoty parametrů U a D se podle kategorií mírně liší od testovacích nahrávek. Pro tichá prostředí T jsou výsledky pro kritérium SAN znatelně horší než pro testovací nahrávky. Mírně si pravidlo $SumMax$ pohoršilo i pro běžná prostředí N a pro rušná prostředí se výsledky mírně zlepšily. Celkově ale nedošlo k velkým výkyvům kvality detektoru řeči pro stanovené parametry a můžeme tak parametry považovat za dostatečně robustní pro použití na libovolných nahrávkách.

6.3.4 Vyhodnocení experimentu

Abych mohl zhodnotit výkon navrženého detektoru řeči pro použité pravidlo $SumMax$ použiji jako referenční hodnotu tabulku výsledků (8) pro výchozí rozhodovací pravidlo Max. Tato reference mi umožňuje porovnání všech používaných kvalitativních ukazatelů.

Při porovnávání budu, oproti předchozím krokům, používat k vyhodnocení mimo úspěšnosti detekce řeči SCR VAD algoritmem vůči referenčnímu signálu a chyby při detekování řeči jako šumu SAN i chybu opačnou NAS spolu s ukazatelem procentuálního výskytu řeči v nahrávce podle rozhodnutí klasifikátoru S_{klas} a podle referenčního signálu S_{ref}.

	U	D	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
T	14	0	96.60	2.05	1.34	16.57	17.53
N	18	2	95.06	0.90	4.02	53.21	42.43
R	22	0	87.52	9.46	3.01	40.24	50.51

Tabulka 9: Referenční tabulka zobrazující výkonové ukazatele detektoru řeči při použití pravidla Max.

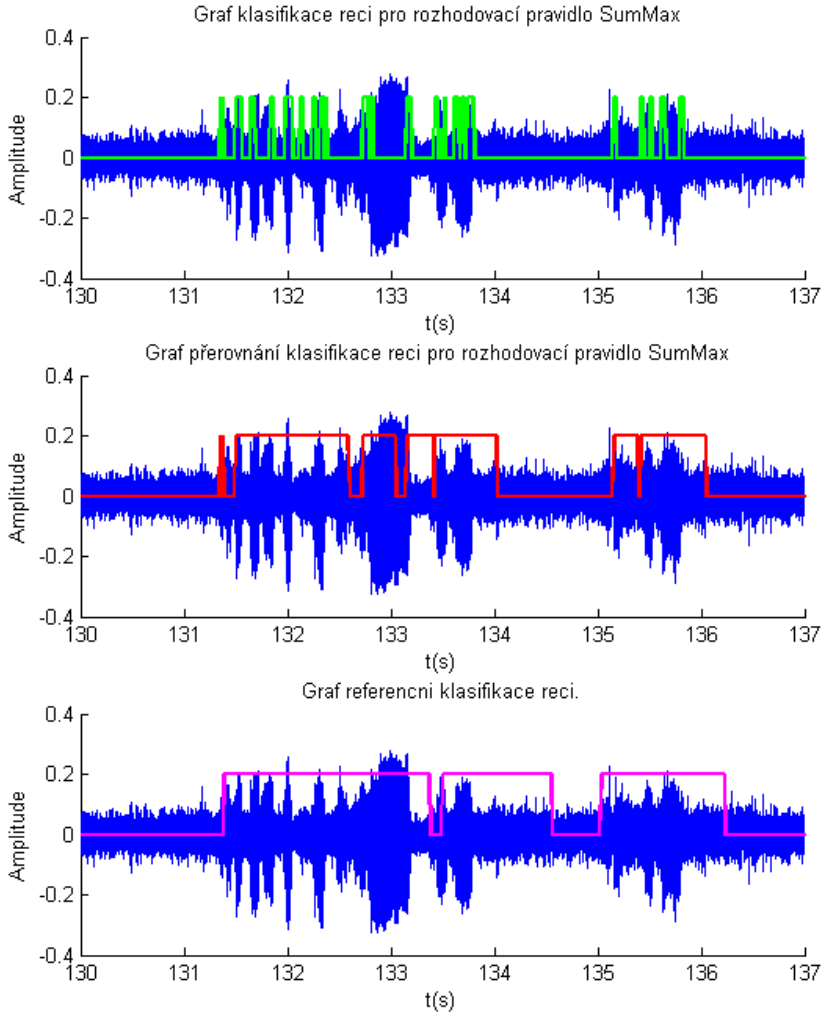
	U	D	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
T	22	0	94.24	4.21	1.54	13.48	17.53
N	6	0	92.44	1.86	5.69	47.38	42.43
R	18	0	88.36	5.52	6.11	49.64	50.51

Tabulka 10: Tabulka zobrazující výkonové ukazatele detektoru řeči při použití pravidla $SumMax$.

Při porovnání výsledků obou rozhodovacích pravidel v tabulkách (9), (10) je vidět zajímavé zlepšení pro vysoce zašuměná akustická prostředí R. Kde oproti použití pravidla Max došlo u výsledků pravidla $SumMax$ k mírnému navýšení úspěšnosti SCR, značnému poklesu chyby typu SAN a zvýšení množství klasifikované řeči S_{klas} až na úroveň referenčního signálu. Nicméně v kontrastu s mnohem vyšším kvalitativním poklesem pro ostaní prostředí, nemohu použití pravidla $SumMax$ v obecném kontextu na místo původního doporučit. Cíl experimentu se tak implementací rozhodovacího pravidla $SumMax$ do algoritmu VAD nenaplnil.

6.3.5 Ukázka průběhu klasikace

Grafická ukázka detekování řeči pravidlem $SumMax$ pro akustické prostředí R.



Obrázek 6: Ukázka VAD pro pravidlo $SumMax$ na akustickém prostředí R

6.4 Experiment 3: Rozhodovací pravidlo $FirstK$

Motivace k dalšímu pokusu přetrvává z předchozího experimentu. Budu se tak opět snažit navýšit kvalitu detekce řeči pro rušná akustická protředí R při zachování dosažených kvalit u ostatních prostředí.

Sumarizace hodnot matice metriky $lMet$ pro jednotlivé Markovské modely se v předchozím experimentu neukázala jako cesta, která by účinně zpřesňovala stanovené ukazatele kvality detekce řeči nezávisle na prostředí. V tomto experimentu zkusím přistoupit k problému z jiného pohledu. Přestože představený natrénovaný akustický model by měl robustně popisovat a zároveň od sebe odlišovat jednotlivé modely fonémů a zvuků okolí, může při použití pravidla Max nastat situace kdy zpracovávaný rámec bude klasifikován podle maximální hodnoty matice $lMet$ do nesprávné třídy, přestože by dalších K stavů HMM s vysokou hodnotou metriky vedlo ke správné klasifikaci.

Pro omezení těchto případů špatné klasifikace rozhodovacím pravidlem opět nahrazuji původní pravidlo maximální pravděpodobnosti Max, tentokrát pravidlem $FirstK$. Toto pravidlo vybírá K nejvyšších hodnot z matice metriky $lMet$. Modely HMM obsahující stavy jejichž hodnoty metriky jsou mezi nejvyššími K hodnotami jsou následně rozděleny do dvou skupin podle toho zda reprezentují řeč či nikoliv. O klasifikaci se rozhodne na základě porovnání relativních četností obou skupin.

Volba parametru K není implicitně známa. Experiment se tak bude skládat nejen z porovnání používaných kvalitativních ukazatelů klasifikačních pravidel Max, $MaxSum$ a $FirstK$ ale také z hledání suboptimální hodnoty tohoto parametru vzhledem ke kvalitě detekce řeči pro všechna definovaná akustická prostředí. Navíc stejně jako v předchozích experimentech se změnou rozhodovacího pravidla vyvstává možnost změny parametrů metody Hangover U a D od již odhadnutých hodnot, z čehož vyplývá, že bude nutné znovu odhad provést.

Navýšení počtu optimalizovaných parametrů ještě více komplikuje situaci při jejich odhadování pro maximálně kvalitní výkon detektoru řeči. Přestože jsou všechny hledané parametry na sobě vzájemně závislé z důvodu časové náročnosti není možné hledat optimální hodnoty společně, navíc pro více akustických prostředí. Musím tak přistoupit k postupnému hledání suboptimálních hodnot jednotlivých parametrů. Jelikož už z předchozích experimentů máme odhady parametrů U a D, můžeme je použít při hledání subop-

timální hodnoty pro K. Konkrétně použiji hodnot nalezených při použití rozhodovacího pravidla Max a to hned ze dvou důvodů. Prvním je větší podobnost pravidla $FirstK$ s pravidlem Max než se $SumMax$. Druhým je pak fakt, že pravidlo Max prozatím stále poskytuje nejlepší výsledky ze zkoušených pravidel. Následně pak při známém K pro každé uvažované akustické prostředí provedu odhad U a D stejně jako v předchozím experimentech.

6.4.1 Odhad parametrů pro akustické prostředí R:

K	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
5	88.63	6.79	4.56	46.63	50.51
10	87.93	5.15	6.90	41.31	50.51
15	87.02	4.43	8.54	51.74	50.51
20	87.00	4.66	8.33	54.51	50.51
25	86.54	4.09	9.35	53.75	50.51
30	86.34	3.48	10.16	55.89	50.51

Tabulka 11: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ pro akustické prostředí R, hodnoty parametrů $U = 22$, $D = 0$ a pro různé hodnoty parametru K .

Jak lze vidět v tabulce (11), pro zvyšující se K se úspěšnost klasifikace SCR snižuje stejně tak jako úroveň chyby typu SAN. To mě opět vede ke zvolení hodnoty K balancováním protichůdných požadavků. Tímto způsobem jsem došel k závěru, že jako ideální volba se jeví zvolit $K = 15$. Při této hodnotě se zachovává uspokojivá úroveň přesnosti SCR, výrazně snižuje chyba typu SAN a klasifikované množství řeči v promluvě nejlépe odpovídá referenčnímu údaji.

Z údajů v tabulce (12) je zřejmé, že nejkvalitnějších výsledků dosahuje testovaný VAD za předpokladu $K = 15$ pro $D = 0$ čili pro nulové zpoždění v klasifikaci v přechodu z úseku ticha na řeč.

Jelikož, po stanovení $K = 15$ a $D = 0$, je parametr U poslední, jehož optimální hodnotu hledám pro rozhodovací pravidlo $FirstK$, výběrem výkonnostně nejlepšího řádku z tabulky (13) tak stanovím hodnoty, které budu porovnávat s údaji ostatních zkoušených rozhodovacích pravidel. Z toho důvodu jsem o tyto údaje tabulku rozšířil na posledních dvou řádkách.

D	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
0	89.50	6.85	3.64	42.75	50.51
2	84.15	13.37	2.46	34.19	50.51
6	70.31	28.64	1.03	17.67	50.51
10	62.92	36.61	0.46	9.50	50.51
14	59.95	39.78	0.26	6.17	50.51
18	57.61	42.20	0.18	3.50	50.51
22	56.73	43.12	0.13	2.50	50.51

Tabulka 12: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ pro akustické prostředí R, pro parametru $K = 15$ a pro různé hodnoty D.

U	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
0	88.28	11.65	0.05	18.31	50.51
2	89.33	10.25	0.40	24.93	50.51
6	89.66	8.27	2.05	35.45	50.51
10	89.50	6.85	3.64	42.75	50.51
14	88.99	5.79	5.21	47.62	50.51
18	88.31	4.96	6.72	51.37	50.51
22	87.02	4.43	8.54	54.51	50.51
Max	87.52	9.46	3.01	40.24	50.51
$SumMax$	88.36	5.52	6.11	49.64	50.51

Tabulka 13: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ pro akustické prostředí R, pro parametru $K = 15$ a pro různé hodnoty U.

Nejvyšší hodnoty úspěšnosti sice dosahuje pravidlo $U = 6$ ale chyba klasifikace typu SAN je pro tuto oblast neúměrně vysoká. Podobně nízké je procentuální zastoupení řeči S_{klas} oproti referenční hodnotě S_{ref}. Při volbě $U = 18$ se sice nepatrně sníží úspěšnost klasifikace ale o více jak třetinu se sníží chyba typu SAN a zastoupení řeči se dostane nad referenční množství. Oproti ostatní možnostem dosahuje pro tuto hodnotu VAD algoritmus nejlepších výsledků.

V porovnání s předešlými pravidly se podařilo pro pravidlo $FirstK$ s parametry $K = 15$, $D = 0$ a $U = 18$ zvýšit kvalitu klasifikace pro rušná akustická prostředí R, obzvláště pak pro chybu typu SAN a procentuální zastoupení řeči nahrávce S_{klas}.

6.4.2 Odhad parametrů pro akustické prostředí T:

K	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
5	96.43	1.46	2.10	19.49	17.53
10	96.12	1.14	2.73	21.35	17.53
15	95.89	1.00	3.09	22.21	17.53
20	96.05	1.17	2.76	21.27	17.53
25	95.87	1.07	3.04	21.98	17.53
30	95.79	0.91	3.28	22.78	17.53

Tabulka 14: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ pro akustické prostředí T, hodnoty parametrů $U = 14$, $D = 0$ a pro různé hodnoty parametru K.

Pro kategorii tichých prostředí T ukazatel úspěšnosti detekce řeči SCR i hodnota chyby typu NAS mírně klesají se zvyšujícím se K ale i tak zůstávají stále vysoké kvalitativní úrovni, viz tabulka (14). Optima soudím, dosahují v bodě kde $K = 10$, za předpokladu nastavení parametrů metody Hangover $U = 14$, $D = 0$, odhadnutých za použití pravidla Max. Nyní přistoupím k upřesnění jejich odhadu pro stávající pravidlo.

D	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
0	96.96	1.52	1.50	18.91	17.53
2	96.60	2.43	0.95	15.94	17.53
6	93.78	5.77	0.43	10.35	17.53
10	90.49	9.16	0.33	5.89	17.53
14	88.54	11.30	0.15	2.94	17.53
18	87.70	12.19	0.10	1.79	17.53
22	87.30	12.58	0.10	1.35	17.53

Tabulka 15: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ pro akustické prostředí T a pro parametru $K = 10$ a pro různé hodnoty D.

V tomto případě je volba optimálního D jednoduchá. Pro tichá prostředí při použití rozhodovacího pravidla $FirstK$ s parametrem $K = 10$ očividně není potřeba používat zpoždění rozhodnutí v metodě Hangover při klasifikaci řeči v období ticha. Očividně toto zpoždění má záporný vliv na kvalitu VAD algoritmu, viz tabulka (15). Proto volím $D = 0$.

U	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
0	96.54	3.31	0.13	8.88	17.53
2	96.89	2.87	0.23	11.64	17.53
6	97.17	2.13	0.68	15.81	17.53
10	96.96	1.52	1.50	18.91	17.53
14	96.12	1.14	2.73	21.35	17.53
18	95.20	0.83	3.96	23.44	17.53
22	94.25	0.66	5.08	25.12	17.53
Max	96.60	2.05	1.34	16.57	17.53
$SumMax$	94.24	4.21	1.54	13.48	17.53

Tabulka 16: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ pro akustické prostředí T a pro parametru $K = 10$ a pro různé hodnoty U.

Při pohledu na hodnoty usuzovaných kritérií pro rozhodovací pravidlo $FirstK$ s parametry $K = 10$ a $D = 0$ v tabulce (16), se může na první pohled zdát jako vhodná volba parametru $U = 6$, protože pro něj je maximální úspěšnost klasifikace a i ostatní parametry se zdají být v normě. Při podrobnější pohledu je ale zřejmé, že pro hodnotu parametru $U = 10$ dosáhne detektor řeči za cenu minimálně snížené celkové úspěšnosti klasifikace významně snížit chybu typu SAN a zároveň se i zvýší zastoupení detekované řeči ve zpracovávané nahrávce lehce nad hodnotu referenční. To je přesně to co potřebuji, protože pokud je hodnota S_{klas} nižší než S_{ref} automaticky je tento rozdíl obsažen v chybě SAN, kterou se snažíme minimalizovat.

V porovnání s dalšími experimentálně testovanými pravidly se pravidlo $FirstK$ s parametry $K = 10$, $D = 0$ a $U = 10$ osvědčilo i pro tichá akustická prostředí T. Při mírném zvýšení už tak relativně vysoké úspěšnosti detekce řeči, dokázalo snížit chybu typu SAN o čtvrtinu a zvýšit procento detekované řeči S_{klas}. Přispělo tedy ke zlepšení téměř ve všech hodnocených kritériích.

6.4.3 Odhad parametrů pro akustické prostředí N:

K	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
5	85.29	0.98	13.71	59.80	42.43
10	83.43	0.53	16.03	63.18	42.43
15	82.19	0.36	17.44	64.95	42.43
20	83.25	0.40	16.34	63.92	42.43
25	81.33	0.35	18.31	66.14	42.43
30	79.81	0.23	19.95	68.04	42.43
Max	95.06	0.90	4.02	53.21	42.43
$SumMax$	86.64	1.62	11.72	55.95	42.43

Tabulka 17: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ pro akustické prostředí N, hodnoty parametrů $U = 18$, $D = 2$ a pro různé hodnoty parametru K.

Pro poslední uvažovanou kategorii běžných prostředí N je ale situace na první pohled jiná. Výsledné hodnoty úspěšnosti SCR jsou oproti dalším pravidlům výrazně nižší pro všechny uvažované hodnoty K. Tento pokles SCR je zapříčiněn vysokým procentuálním výskytem „řeči“ detekované VAD algoritmem v nahrávce S_{klas}. To má za následek i výrazný rozdíl mezi chybou typu NAS pro pravidlo Max a $FirstK$. Optimální hodnotu pro parametr K z množiny testovaných hodnot nemá smysl, protože žádná z nich neposkytuje dostatečně kvalitní informaci o výskytu nahrávky v promluvě.

Zde je nutné provést úvahu, že při volbě $K = 1$ se vybere pouze maximální hodnota z matice metriky $lMet$ na základě, které dochází ke klasifikaci řeči či šumu podle toho kterému HMM vybraná hodnota náleží. To odpovídá popisu funkce pravidla Max a z toho vyplývá, že při volbě parametru $K = 1$ pravidlo $FirstK$ je degradováno na pravidlo Max. Zároveň s tím je vhodné si uvědomit, že zatím nejlepší výsledky pro nahrávku náležící do kategorie N jsme získali právě použitím pravidla Max.

Dále tak i odpadá nutnost optimalizovat parametry metody Hangover, protože už tak bylo učiněno v prvním experimentu a hodnoty v tabulce jsou vypočteny právě za jejich použití čili s $U = 18$ a $D = 2$.

6.4.4 Ověření výsledků

Pokud uvážím závěry diskutované pro pravidlo $FirstK$ na všech uvažovaných kategoriích akustických prostředí uvedených v předchozích podkapitolách, lze tvrdit, že pravidlo $FirstK$ poskytuje pro specifické hodnoty parametru K stejně kvalitní či dokonce kvalitnější výsledky při detekci řeči než je tomu za použití původního pravidla Max. To zatím ale platí pouze na nastavovacích nahrávkách. Abych toto mohl tvrdit obecně, je nutné ověřit funkci této konfigurace VAD algoritmu na testovacích nahrávkách pro každé akustické prostředí.

	K	SCR [%]	SAN [%]	NAS [%]	S_{klas}[%]	S_{ref}[%]
T	10	96.51	1.68	1.79	26.32	25.32
N	1	91.96	1.60	6.43	34.96	40.93
R	15	90.29	6.57	3.13	39.06	45.56

Tabulka 18: Tabulka zobrazující výkonové ukazatele VAD při použití pravidla $FirstK$ s optimálně zvoleným parametrem K pro testovací nahrávky ze všech akustických prostředí.

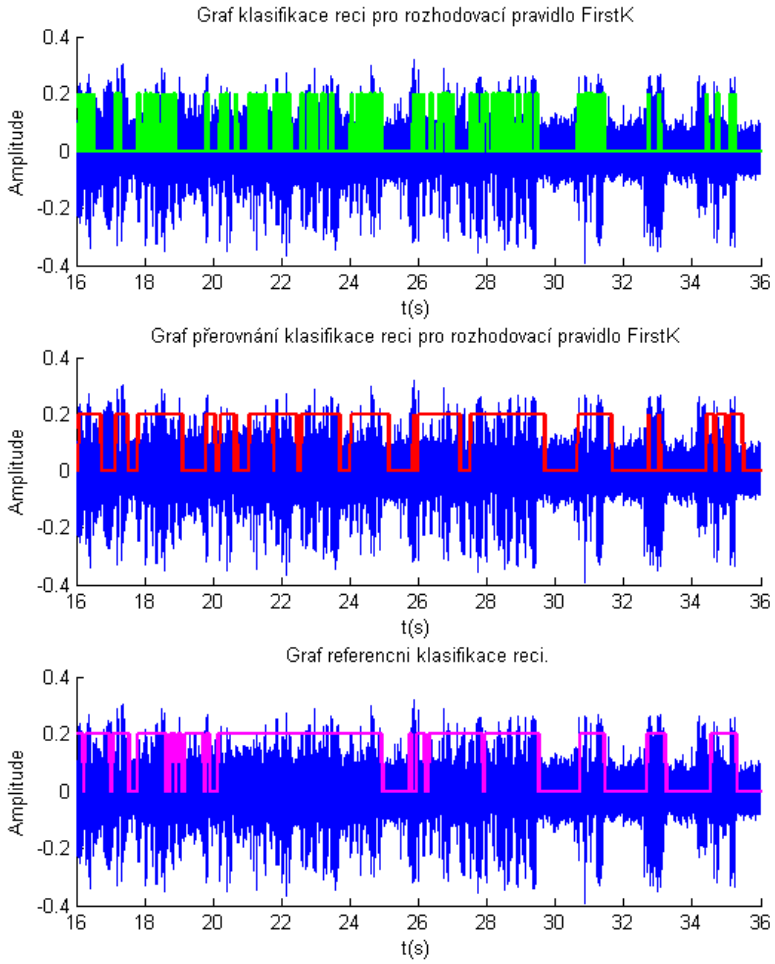
Z přehledu ukazatelů kvality v tabulce (18), je zřejmé, že ověření mohu považovat za splněné, protože kvalita detekce řeči za použití rozhodovacího pravidla $FirstK$ zůstává na testovacích nahrávkách pro všechna prostředí stabilně vysoká. Pro tiché prostředí T jsou výsledky téměř beze změny pro všechny ukazatele. U kategorie běžných prostředí došlo k mírnému poklesu ale pouze minimálně. Naopak u rušných prostředí R se úspěšnost detekce řeči ještě zvýšila i přes mírný nárůst chyby typu SAN.

6.4.5 Vyhodnocení experimentu

V tomto experimentu jsem se snažil nahrazením původního rozhodovacího pravidla Max pravidlem $FirstK$ s parametrem K minimálně zachovat, lépe zvýšit výkon a kvalitu navrženého VAD algoritmu pro libovolné akustické prostředí. Zvýšit výkon VAD algoritmu se mi podařilo pro tiché a rušné prostředí T, R. Kdy pro obě jsem byl schopen určit specifické suboptimální hodnoty parametrů U, D metody vyhlazení výstupu klasifikátoru Hangover a K pravidla $FirstK$, tak aby VAD poskytoval kvalitnější výsledky než za použití pravidla Max. V případě běžných akustických prostředí jsem dokázal zachovat původní výkon, zjednodušením funkce pravidla $FirstK$ volbou parametru $K = 1$. Což mělo za následek jeho zjednodušení do podoby pravidla Max. Čímž i byla splněna motivace tohoto experimentu.

6.4.6 Ukázka průběhu klasikace

Grafická ukázka detekování řeči pravidlem $FirstK$ s parametrem $K = 15$ pro testovací nahrávku akustického prostředí R.



Obrázek 7: Ukázka VAD pro pravidlo $FirstK$ pro $K = 15$ na testovací nahrávce v akustickém prostředí R.

7 Závěr

Tato práce je motivována snahou vyvinout robustní algoritmus pro stále nevyřešenou problematiku detekování řeči, kde by především pro nestacionární či silně zarušené akustické prostředí nedocházelo k přílišným poklesům kvality indikace řeči.

V počátku práce jsem se věnoval shromáždění a nastudování podkladů nutných pro seznámení se s historickým vývojem a současným stavem problematiky detekce řeči. Konkrétně jsem se zaměřil na různé druhy přístupů uplatňovaných při realizaci VAD algoritmů, jejich vlastnosti a vhodnost použití ve specifických prostředích.

Následně jsem provedl teoretickou analýzu vybraných metod založených na principech prahové detekce, statistického přístupu či použití dlouhodobé informace.

Jako nejvhodnější z nich jsem pro návrh vlastního detektoru řeči, s ohledem na reálné použití v aplikaci automatického titulkování televizních pořadů, vyhodnotil právě statistický přístup, jehož se využívá i v navazujícím procesu rozpoznávání řeči. Nabízí se tak možnost využít pro obě úlohy společné rekvizity, jako je například v mém případě vytvořený a natrénovaný akustický model.

Ten je vytvořen sloučením dvou submodelů složených z množiny monofónních pětistavových „levo-pravých“ skrytých Markovských modelů, viz kapitola 4.2.2. Přičemž submodel řeči popisuje jednotlivými HMM fóny českého jazyka a submodel šumu různé zvuky. Tyto submodely jsou před sloučením natrénovány na oddělených trénovacích datech.

Za trénovací sadu pro model řeči jsem použil nahrávky zapůjčené vedoucím studia. Naproti tomu jsem, ve shodě se zadáním, trénovací data modelu šumu získal ze zdroje [10], kde jsem z volně dostupných zdrojů vybral reprezentativní množinu zvuků objevujících se ve třech stanovených kategoriích akustických prostředí (tiché, běžné, rušné). Poté bylo nutné sjednotit formu těchto nahrávek převedením do stejného audio formátu a převzorkováním na shodnou vzorkovací frekvenci.

Testovací sada nahrávek odpovídá zvolené aplikaci a má tak povahu záznamů úseků televizných přenosů z různých akustických prostředí aby odpovídali stanoveným kategoriím.

Jako vhodnou reprezentaci nahrávek jsem pro následné zpracování jsem určil MFCC koeficienty, pro jejich schopnost robustně popsat spektrální řečové charakteristiky a zanedbatelnou vzájemnou korelovanost.

Pro manipulaci s akustickými modely jsem použil sadu nástrojů HTK, která je zaměřena na práci s HMM a mimo jiné obsahuje nástroje pro parametrizaci nahrávek a trénování akustického modelu.

S takto předpřipravenými daty jsem přistoupil k realizaci algoritmu detekce řeči v programovacím jazyce Java, skrze vývojové prostředí Eclipse.

Dále provádím obecnou úvahu, že na detektor řeči může být nahlíženo jako na klasifikátor, který podle vypočtené metriky klasifikuje vstupní data do dvou tříd. Přičemž, základní ideou je, že klasifikátor je schopen na základě zvoleného rozhodovacího pravidla, rozlišit řečový signál od libovolného šumu v pozadí.

Jako vhodnou informaci neboli metriku pro klasifikaci rámců jsem vybral výpočet podmíněné pravděpodobnosti generování vektoru MFCC koeficientů zpracovávaného rámce emitujícím stavem skrytého Markovského modelu.

K vyhodnocení kvality detekce řeči používám čtyři ukazatele, úspěšnost SCR, chyby typu SAN a NAS, relativní množství detekované řeči S_{klas}, definované v kapitole 4.6. Tyto ukazatele jsou vypočteny porovnáním výsledků klasifikace se správnou referenční hodnotou s mírou tolerance odpovídající $0,1[s]$ (míra tolerance $T = 10$).

Dále se mi pomocí experimentů, popsaných v kapitolách 6.2, 6.3, 6.4, podařilo pro všechna zkoušená pravidla a akustická prostředí odhadnout a následně ověřit optimální hodnoty dvojice parametrů metody Hangover, jež vyhlazuje data na výstupu klasifikátoru, ve smyslu snížení fluktuace klasifikace v pozvolných přechodech mezi řečí a šumem a naopak. To se realizuje úmyslně zaneseným zpožděním změny klasifikace ve zmíněných přechodech, kde nalezené hodnoty parametrů metody představují právě velikost těchto zpoždění (U: řeč $\rightarrow$ šum, D: šum $\rightarrow$ řeč). Zajímavým zjištěním bylo, že, nezáleže na akustickém prostředí, úspěšnost detekce řeči téměř vždy rostla se snižující se hodnotou parametru D až k nule a naopak často rostla pro zvyšující se U, blíže například v kapitole 6.2.4.

Při návrhu VAD jsem za výchozí volbu rozhodovacího pravidla stanovil pravidlo maximální pravděpodobnosti Max. Pro optimální hodnoty parametrů metody Hangover vzhledem k prostředí, poskytoval VAD algoritmus s pravidlem Max uspokojivě kvalitní výsledky pro tichá T a běžná N akustická prostředí. U rušných prostředí R byl ale patrný znatelný pokles.

Z toho důvodu jsem přistoupil k navazujícím experimentům kde jsem se snažil při zachování stávající vysoké úspěšnosti u tichých a běžných prostředí dosáhnout vyšší kvality VAD pro prostředí rušná, záměnou pravidla Max za pravidla $SumMax$(6.3) a $FirstK$(6.4).

Pro pravidlo $SumMax$ se mi povedlo dosáhnout mírně kvalitnějších výsledků pro rušná prostředí ale za cenu výrazného poklesu kvality pro ta zbylá. Další pravidlo $FirstK$ obsahuje parametr K, které zjednodušeně označuje počet nejvyšších hodnot z vektoru metriky, které se budou pro klasifikaci používat. Experimentováním s tímto parametrem se mi skutečně podařilo dosáhnout buď stejně kvalitních nebo kvalitnějších výsledků pro všechna uva-

žovaná prostředí. Po ověření těchto závěrů na ověřovací nahrávce testovací sady, mohu původně použité rozhodovací pravidlo v algoritmu VAD zaměnit za pravidlo $FirstK$ a prohlásit algoritmus VAD za do jisté míry robustní vůči volbě akustického prostředí.

Seznam obrázků

Seznam tabulek

8 Přílohy

8.1 Příloha 1: algoritmus pro vyhlazení finální klasifikace

```
U = 5;
D = 10;
counterU = 0;
counterD = 0;
for iFrame = 1 : nFrame
   if FrameClassFinal[iFrame-1] == 1
      //pokud je řeč, bude řečí
      if FrameClassFirst[iFrame] == 1
         FrameClassFinal[iFrame] = 1;
         counterU = 0;
      else
      //pokud je ticho, záleží kolikátý
      //rámec ticha v řadě to je
         if counterU < U
            counterU++;
            FrameClassFinal[iFrame] = 1;
        else
            FrameClassFinal[iFrame] = 0;
         end
      end
   else
      //pokud je ticho, bude tichem
      if FrameClassFirst[iFrame] == 0
         FrameClassFinal[iFrame] = 0;
         counterD = 0;
      else
      //pokud je rec, záleží kolikátá
      //řeč v řadě to je
         if counterD < D
            counterD++;
            FrameClassFinal[iFrame] = 0;
         else
            for(int i = 0; i < D+1; i++){
               FrameClassFinal[iFrame-i] = 1;
            end
         end
      end
```

```
   end
end
```

Vysvětlivky:

- iFrame: index zpracovávaného rámce
- U: počet rámců, při zpoždění klasifikace v přechodu z řeči do ticha
- D: počet rámců, při zpoždění klasifikace v přechodu z ticha do řeči
- counterU: počítadlo rámců ticha při přechodu
- counterD: počítadlo rámců řeči při přechodu
- FrameClassFirst: původní klasifikace
- FrameClassFinal: finální klasifikace

8.2 Příloha 2: algoritmus pro výpočet úspěšnosti SCR a chyb typu SAN a NAS

```
err = false;
if Ref[i] == Test[i]
   SCR++;
   last = Ref[i];
else
   find = false;
   if last == 1
      for j = 1 : T+1
         if Ref[i-j] == Test[i-j] && Test[i-j] == 1
            SCR++;
            find = true;
            break;
         end
      end
      if !find
         err = true;
      end
   else
      for j = 1 : T+1
         if Ref[i+j] == Test[i+j] && Test[i+j] == 1
            SCR++;
            find = true;
            break;
         end
      end
      if(!find) //najít další shodu
         next = 1;
         val = false;
         for j = 1 : T+1
            if Ref[i+j] == Test[i+j]
               next = Ref[i+j];
               val = true;
               break;
            end
         end
         if next == 0
            for j = 1 : T+1
               if Ref[i+j] == Test[i+j] && Test[i+j] == 0
                  SCR++;
                  find = true;
```

```
                break;
            end
         end
         if(!find){//chyba
            err = true;
         end
      else
         err = true;//chyba
      end
   end
end
if err
   if Ref[i] == 1
   //řeč klasifikovaná jako šum
      SAN++;
   else
   //šum klasifikovaný jako řeč
      NAS++;
   end
end
```

Vysvětlivky:

- i: index zpracovávaného rámce
- Ref: pole obsahující správnou klasifikaci rámců podle reference
- Test: pole obsahující klasifikaci rámců podle navrženého VAD algoritmu
- last: hodnota, již nabývala poslední shodující se klasifikace
- T: míra tolerance