LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTIKAS NODAĻA

SLĒPTO MARKOVA MODEĻU SISTĒMAS IZSTRĀDE RUNĀTO CIPARU ATPAZĪŠANAI, IZMANTOJOT MFCC PAZĪMES.

KURSA DARBS

Autors: **Kirills Bobkovs**
Stud. apl. kbxxxxx
Darba vadītājs: prof. Dr.math. Uldis Strautiņš

RĪGA 2025

Anotācija

Šis kursa darbs ir veltīts runas apstrādei, izmantojot Fourier transformāciju un Mel filtru bankas, un pēc tam koncentrējas uz mašīnmācīšanās sistēmas izstrādi un īstenošanu runāto ciparu (no 1 līdz 9) atpazīšanai, izmantojot Slēptos Markova modeļus (SMM). Sistēma izmanto Mel-frekvences cepstrālos koeficientus (MFCC), lai izvilktu iezīmes no audio ierakstiem, kas kalpo kā galvenie ievades dati modeļa apmācībai un klasifikācijai. Sistēma tiek apmācīta, izmantojot 2940 runātus ciparus no 6 dažādiem runātājiem, katrs runātājs katru ciparu izrunājot 49 reizes. Šie audio faili tiek izmantoti, lai veidotu Gausa SMM, kas tiek izmantoti, lai modelētu runas temporālās un spektrālās īpašības. Apmācītie modeļi tiek glabāti turpmākai lietošanai un var tikt dinamiski ielādēti, nodrošinot aprēķinu efektivitāti testēšanas laikā.

Testēšanas posmā modelis tiek novērtēts, izmantojot atsevišķu 60 audio failu kopu no 6 dažādiem runātājiem, kopumā 60 testēšanas paraugus. Apmācītā modeļa atpazīšanas precizitāte ir 91,67%. Šī augstā veiktspēja norāda, ka Slēptie Markova modeļi, apvienojot tos ar MFCC iezīmju izvilkšanu, nodrošina efektīvu pieeju runas atpazīšanai šajā kontekstā. Turklāt matemātiskās pamatprincipi, kas ir būtiski runas apstrādē, piemēram, Fourier transformācija un Mel filtru banka, bija svarīgi, lai izvilktu attiecīgās iezīmes. Pētījums arī uzsver SMM pielietojumu, lai novērtētu stāvokļu pārejas un emisijas varbūtības, izmantojot algoritmus, piemēram, uz priekšu-atpakaļ metodi varbūtības novērtēšanai.

Rezultāti liecina par sistēmas spēju klasificēt runātos ciparus ar augstu precizitāti, tomēr modeļa veiktspēju var ietekmēt līdzība starp apmācības un testēšanas datu kopām. Neskatoties uz to, ka datu kopas ir savstarpēji nesakritīgas, abi datu kopumi tika ierakstīti no tiem pašiem runātājiem un vienādos apstākļos, kas var novest pie noteikta līmeņa ”homogenitātes” datos. Tas varētu veicināt novēroto augsto precizitāti, taču arī radīt šaubas par to, cik labi modelis varētu vispārināt jaunus, neredzētus runātājus vai dažādākus ierakstu apstākļus. Tādējādi modeļa veiktspēja, iespējams, var mainīties atkarībā no datu homogenizācijas līmeņa. Paplašinot datu kopu, iekļaujot daudzveidīgāku runātāju un ierakstu apstākļu diapazonu, kā arī izpētot alternatīvas statistiskās modelēšanas tehnikas, varētu palīdzēt uzlabot vispārināšanu un noturību.

Atslēgas vārdi: Runas atpazīšana, Furjē transformācija, Mel-frekvences cepstrālie koeficienti, Slēptie Markova modeļi

Abstract

This coursework is aimed at processing speech using Fourier Transform and Mel filter banks and subsequently focuses on the design and implementation of a machine learning system for the recognition of spoken digits (from 1 to 9) using Hidden Markov Models (HMM). The system leverages Mel-Frequency Cepstral Coefficients (MFCCs) to extract features from audio recordings, which serve as the primary input for model training and classification. The system is trained using a set of 2940 spoken numbers by 6 different speakers, with each speaker pronouncing each digit 49 times. These audio files are used to build Gaussian HMMs, which are then employed to model the temporal and spectral patterns of speech. The trained models are stored for later use and can be loaded dynamically, ensuring computational efficiency during testing

During the testing phase, the model is evaluated using a separate set of 60 audio files from 6 different speakers, totaling 60 test samples. The recognition accuracy achieved by the trained model is 91.67%. This high performance indicates that Hidden Markov Models, combined with MFCC feature extraction, provide an effective approach to speech recognition in this context. Furthermore, the mathematical foundations underpinning speech processing, such as the Fourier Transform and the Mel filter bank, were crucial for extracting relevant features. The study also highlights the application of HMMs in estimating state transition and emission probabilities, using algorithms like the forward-backward method for likelihood estimation.

The results demonstrate the system's ability to classify spoken digits with high accuracy, though the performance of the model may be influenced by the similarity between the training and testing datasets. Despite being disjoint, both datasets were recorded from the same speakers and under the same conditions, which could lead to a certain level of 'homogeneity' in the data. This could contribute to the high accuracy observed, but also raise doubts about how well the model would generalize to new, unseen speakers or more varied recording environments. Therefore, the model's performance can likely vary depending on the level of homogenization in the data. Expanding the dataset to include a more diverse range of speakers and recording conditions, as well as exploring alternative statistical modeling techniques, could help improve generalization and robustness.

Keywords: Speech Recognition,Fourier transform, Mel-Frequency Cepstral Coefficients, Hidden Markov Models

Saturs

Apzīmejumi

A Pārejas matrica

B Emisijas matrica

π Sākotnējo stāvokļu sadalījums

$\mathcal{L}$ Mācīšanās funkcija, kas tiek izmantota HMM apmācībā, piemēram, log-likelihood funkcija.

$f_{\textbf{mel}}$ Mel frekvence

c_n Mel-frekvences cepstrālais koeficients (MFCC)

Δx Laika dispersija

$\Delta \xi$ Frekvences dispersija

$X(f)$ Furjē transformācija signāla $x(t)$

$w(n)$ Loga funkcija

$f_{\textbf{Nyquist}}$ Nyquist frekvence

Ievads

Runas atpazīšana ir starpdisciplināra datorzinātnes un datorlingvistikas apakšnozare, kas ietver metodoloģijas un tehnoloģijas, kas ļauj datoriem atpazīt un tulkot runāto valodu tekstā. Pilnīgs risinājums joprojām ir tālu no sasniedzamā mērķa. Runas atpazīšanas tehnikas ietver Dabas valodas apstrādi (NLP), kas koncentrējas uz cilvēka un mašīnas mijiedarbību caur runu un tekstu, un bieži tiek izmantota balss meklēšanas lietojumprogrammās. N-grammas, kas ir vienkāršākais valodas modelis, piešķir varbūtības vārdu secībām, uzlabojot atpazīšanu, izmantojot gramatiku un vārdu secību varbūtības. Neironu tīkli, ko izmanto dziļās mācīšanās algoritmos, apstrādā datus, atdarinot cilvēka smadzeņu savienojamību, piedāvājot lielāku precizitāti, taču ar lēnāku mācīšanās procesu. Turklāt runātāja diarizācijas algoritmi palīdz identificēt un segmentēt runu pēc runātāju identitātes, bieži izmantojot zvanu centros.

Slēptie Markova modeļi (SMM) ir probabilistiskie modeļi, kas tiek izmantoti secību modeļu izveidošanai, kas piešķir etiķetes katrai vienībai secībā, piemēram, vārdiem, zilbēm vai teikumiem. Šīs etiķetes veido kartējumu ar ievadīto informāciju, ļaujot modelim noteikt vispiemērotāko etiķešu secību. SMM izmanto slēptos stāvokļus, kas nav tieši novērojami, bet tiek secināti no novērotajiem datiem, piemēram, runas signāliem. Šajā darbā slēptie stāvokļi tiek uzskatīti par etiķetēm, kas jānosaka, un uzdevums ir precīzi noteikt slēpto stāvokļu secību, kas vislabāk atbilst ievades datiem.

…

1 Teorijas apskats

1.1 Skaņas apstrāde

1.1.1 Furjē transformācija

Furjē transformācija ir matemātisks rīks, kas sadala signālu tā veidojošajās frekvencēs. Nepārtrauktam signālam $x(t)$ Furjē transformācija $X(f)$ tiek definēta kā:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft}\, dt,$$

kur:

- $x(t)$ ir laika domēna signāls,
- $X(f)$ ir frekvenču domēna reprezentācija,
- f ir frekvence hercos (Hz),
- j ir imaginārā vienība ($j^2 = -1$).

Diskrētiem signāliem tiek izmantota Diskrētā Furjē transformācija (DFT):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn},$$

kur:

- $x[n]$ ir diskrēta laika signāls,
- $X[k]$ ir frekvenču domēna reprezentācija frekvenču joslā k,
- N ir paraugu skaits,
- k ir frekvenču joslas indekss ($k = 0, 1, \ldots, N-1$).

Ātrā Furjē transformācija (FFT) ir efektīvs algoritms DFT aprēķināšanai.

1.1.2 Mel filtri

Mel skala ir perceptuāla skala, kas balstīta uz cilvēka auss nelineāro frekvenču izšķirtspēju. Mel frekvence m noteiktai frekvencei f hercos tiek aprēķināta kā:

$$m = 2595 \log_{10} \left(1 + \frac{f}{700}\right).$$

Mel filtri ir trīsstūrveida filtri, kas tiek piemēroti signāla jaudas spektram, lai simulētu cilvēka dzirdes sistēmas reakciju. Šie filtri ir vienmērīgi izvietoti zemās frekvencēs un logaritmiski augstās frekvencēs. Mel filtru izeja ir Mel filtrēts spektrogramma, kas uzsver perceptuāli nozīmīgās frekvences.

1.1.3 Viļņforma

Viļņforma ir laika domēna signāla reprezentācija, kas parasti tiek paraugota ar fiksētu frekvenci (piemēram, 16 kHz). Matemātiski viļņformu var aprakstīt kā funkciju $x(t)$, kur t ir laika mainīgais. Diskrētā laika signālu $x[n]$ iegūst, izmantojot paraugu ņemšanu no nepārtrauktā signāla $x(t)$ ar fiksētu intervālu Δt:

$$x[n] = x(n\Delta t),$$

kur n ir parauga indekss, un Δt ir paraugu ņemšanas periods. Viļņforma atspoguļo signāla amplitūdas izmaiņas laikā.

Mērķis: skaņas analīze un atpazīšana

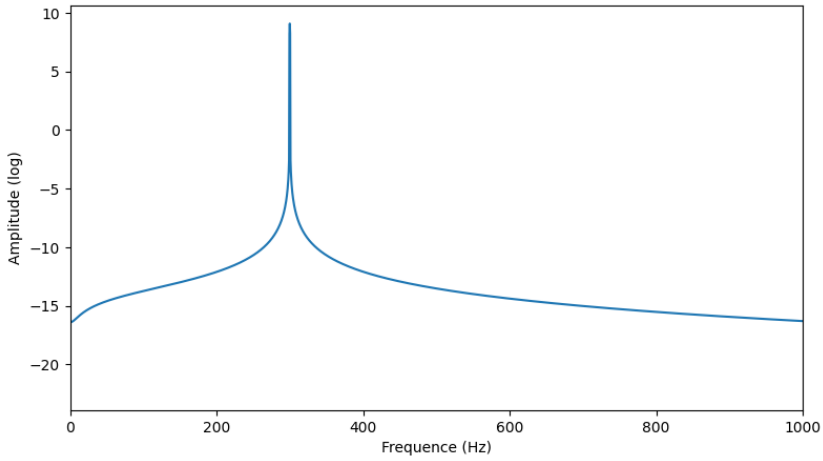
Mūsu mērķis ir analizēt un atpazīt skaņas, izmantojot tās skaitlisko vai grafisko reprezentāciju. Skaņas signāli ir sarežģīti un satur daudz informācijas, piemēram, frekvences, amplitūdas un fāzes. Lai saprastu šo informāciju, mēs pārveidojam laika domēna signālu (viļņformu) frekvenču domēnā, izmantojot Furjē transformāciju. Šī transformācija ļauj iegūt signāla spektru, kas atspoguļo enerģijas sadalījumu pa frekvencēm.

1.1.4 Spektrs

Spektrs tiek definēts kā Furjē transformācijas absolūtā vērtība:

$$|X(f)| = \sqrt{\text{Re}(X(f))^2 + \text{Im}(X(f))^2},$$

kur $X(f)$ ir Furjē transformācijas rezultāts, $\text{Re}(X(f))$ un $\text{Im}(X(f))$ ir attiecīgi reālā un imaginārā daļa. Spektrs $|X(f)|$ parāda signāla amplitūdu katrā frekvencē f.



Att. 1.1: 300 Hz sinusoidālās skaņas spektrs

Par jaudas spektru sauc:

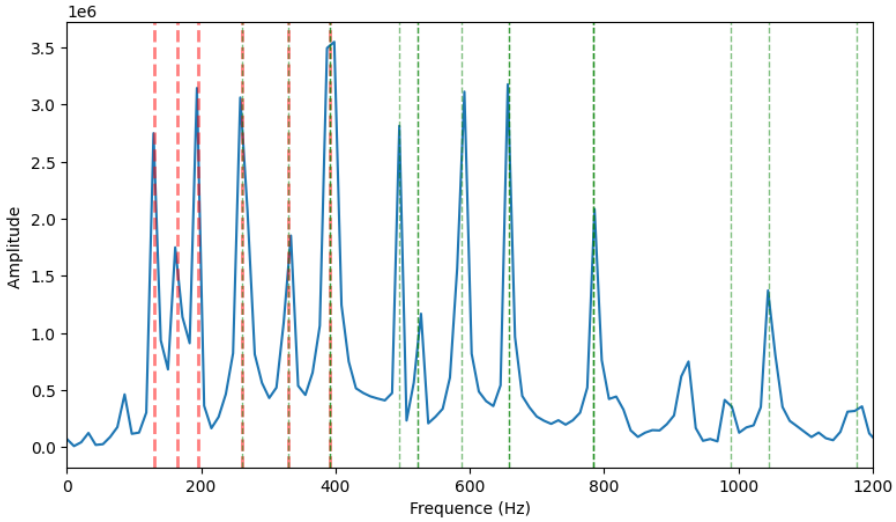
$$P(f) = |X(f)|^2,$$

Šī informācija ir būtiska, lai saprastu signāla frekvenču saturu, taču tā nepietiek, lai analizētu signāla izmaiņas laikā. Tāpēc mēs izmantojam spektrogrammu, kas ir laika-frekvences reprezentācija, kas ļauj vizualizēt, kā signāla frekvences mainās laika gaitā.

1.1.5 Spektrogramma

Spektrogramma ir laika-frekvences reprezentācija, kas iegūta, sakārtojot secīgu rāmju jaudas spektrus. Diskrēta gadījumā tā ir 2D matrica, kur:

- x-ass (kolonas) atspoguļo laiku,
- y-ass (rindas) atspoguļo frekvenci,
- vērtības matricā atspoguļo enerģiju $P[k]$ vai $|X(f)|$.



Att. 1.2: Spektrs. C (Do) mažora akords uz ģitāras. Sarkanas vertikālās līnijas norāda akorda notu frekvences, bet zaļās līnijas norāda virsskaņas.

1.1.6 Īslaicīgā Furjē transformācija (STFT)

Īslaicīgā Furjē transformācija (STFT) ir formālā metode, kas realizē pārveidošanu no viļņformas uz spektrogrammu. STFT sadala signālu īsos laika rāmjos un katram rāmim piemēro Furjē transformāciju, lai iegūtu laika-frekvences reprezentāciju.

$$X[m,k] = \sum_{n=0}^{N-1} x[n]w[n-mH]e^{-j\frac{2\pi}{N}kn},$$

kur:

- $x[n]$ ir diskrēta laika signāls,
- $w[n]$ ir loga funkcija (piemēram, Hamming logs),
- m ir rāmja indekss,
- k ir frekvenču joslas indekss,
- N ir FFT punktu skaits (nfft),
- H ir solis starp rāmjiem (hop_length or frame shift).

1.1.7 Loga funkcija (Hamming logs)

Lai samazinātu spektrālās noplūdes, signāls tiek reizināts ar loga funkciju pirms Furjē transformācijas. Loga funkcijas mērķis ir samazināt signāla pēkšņās robežas, kas citādi izraisītu spektrālo noplūdi — enerģijas izkliedi pa blakus esošajām frekvencēm.

Vispārīga Hamming loga funkcija ir definēta kā:

$$w[n] = \alpha - (1 - \alpha) \cos\left(\frac{2\pi n}{N-1}\right), \quad n = 0, 1, \ldots, N-1,$$

kur:

- $w[n]$ ir loga funkcijas vērtība paraugā n,
- α ir svara koeficients, kas nosaka loga funkcijas formu,
- N ir loga funkcijas izmērs (paraugu skaits).

Hamming logs ir īpašs Hamming loga funkcijas gadījums, kad $\alpha = 0.54$. Šajā gadījumā loga funkcija ir definēta kā:

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right).$$

Šī funkcija ir optimizēta, lai samazinātu blakus esošo frekvenču ietekmi (sānu lobu līmeni), vienlaikus saglabājot pietiekami šauru galveno lobu. Tas padara Hamming logu piemērotu lietošanai audio un signālu apstrādē, kur nepieciešams precīzs frekvenču atdalījums [1].

Hamming logs efektīvi samazina spektrālo noplūdi, vājinot signāla robežas un tādējādi samazinot pēkšņās pārejas starp rāmjiem. Tas ļauj iegūt precīzāku frekvenču analīzi, kas ir būtiska daudzos signālu apstrādes uzdevumos.

1.1.8 Priekšuzsvars

Priekšuzsvars ir augsto frekvenču pastiprināšana, kas tiek piemērota signālam pirms STFT. Tā palīdz uzlabot augsto frekvenču komponenšu attēlojumu spektrogrammā. Preemfāzes filtra formula ir:

$$x_{\text{pre}}[n] = x[n] - \alpha x[n-1],$$

kur α ir priekšuzsvars koeficients (parasti 0.97).

1.1.9 Nyquist frekvence un aliasing efekts

Saskaņā ar Nyquist teorēmu, lai precīzi atjaunotu nepārtrauktu signālu no tā paraugiem, paraugu frekvencei f_s jābūt vismaz divreiz lielākai par signāla augstāko frekvenci f_{max}. Nyquist frekvence f_{Nyquist} ir definēta kā:

$$f_{\text{Nyquist}} = \frac{f_s}{2}.$$

Piemēram, ja paraugu frekvence ir 16 kHz, Nyquist frekvence ir 8 kHz. Frekvences, kas pārsniedz Nyquist frekvenci, tiek nepareizi atspoguļotas spektrā kā zemākas frekvences, izraisot tā saukto aliasing efektu.

Divas frekvences f_1 un f_2 tiek sauktas par savstarpēji aizstājamām frekvencēm (alias frekvencēm), ja eksistē vesels skaitlis $k \in \mathbb{Z}$, ka:[2]:

$$f_2 = f_1 + kf_s,$$

kur:

- f_1 un f_2 ir frekvences, kas rada identiskus paraugus,
- k ir vesels skaitlis ($k \in \mathbb{Z}$),
- f_s ir paraugu frekvence.

Ja signāls tiek paraugots ar frekvenci f_s, divi viļņi ar frekvencēm f_1 un f_2 radīs identiskus paraugus:

$$\cos(2\pi f_1 t + \phi) \quad \text{un} \quad \cos(2\pi f_2 t + \phi),$$

kur $f_2 = f_1 + kf_s$. Šī iemesla dēļ frekvences f_1 un f_2 nevar atšķirt no paraugiem, kas izraisa aliasing.

Lai novērstu aliasing, pirms paraugu ņemšanas tiek izmantots zemas caurlaidības filtrs, kas noņem visas frekvences, kas pārsniedz Nyquist frekvenci. Šī filtra izmantošana ir būtiska, lai nodrošinātu precīzu signāla atjaunošanu no tā paraugiem.

1.1.10 STFT analīze

STFT rezultāts ir 2D kompleksā matrica $X[m, k]$, kur:

- m ir laika rāmja indekss ($m = 0, 1, \ldots, M-1$),
- k ir frekvenču joslas indekss ($k = 0, 1, \ldots, K-1$).

Parametri un izšķirtspēja

STFT parametri un to ietekme:

- **n_fft** (loga garums):
  - Definēts kā Furje transformācijas loga garums, kas parasti tiek izvēlēts kā $n_{\text{fft}} = 2^p$, kur p ir vesels skaitlis, lai nodrošinātu efektīvu FFT aprēķinu. Praksē bieži izmanto 25 ms logu, kas atbilst $n_{\text{fft}} = \lfloor 0.025 \cdot f_s \rfloor$, kur f_s ir paraugu frekvence.
  - Nosaka maksimālo frekvenču izšķirtspēju: $\Delta f = \frac{f_s}{n_{\text{fft}}}$.
  - Lielāks n_{fft} palielina frekvenču izšķirtspēju ($\downarrow \Delta f$), bet samazina laika izšķirtspēju.
- **hop_length** (solis starp rāmjiem):
  - Definēts kā nobīde starp blakus esošo logu sākumiem. Parasti tiek izvēlēts kā hop_length $= \lfloor 0.01 \cdot f_s \rfloor$, kas nodrošina 15 ms pārklājumu, ja izmanto 25 ms logu.
  - Nosaka laika izšķirtspēju: $\Delta t = \frac{\text{hop\_length}}{f_s}$.
  - Mazāks solis (hop_length) palielina laika izšķirtspēju ($\downarrow \Delta t$), bet rada vairāk pārklājuma.

Frekvenču diapazons un sadalījums

Frekvenču asis īpašības:

- **Diapazons**: No 0 līdz Nyquist frekvencei $f_{\text{Nyquist}} = \frac{f_s}{2}$.
- **Frekvenču joslas (frequency bins)**:

$$f[k] = k \cdot \Delta f = k \cdot \frac{f_s}{n_{\text{fft}}}, \quad k = 0, 1, \ldots, \left\lfloor \frac{n_{\text{fft}}}{2} \right\rfloor$$

  Katra frekvenču josla (frequency bin) atspoguļo frekvenču intervālu $[f[k], f[k+1]]$.
- **Frekvenču joslu skaits**: $K = \left\lfloor \frac{n_{\text{fft}}}{2} \right\rfloor + 1$.

Laika asis īpašības

- **Laika momenti (time points)**:

$$t[m] = m \cdot \Delta t = m \cdot \frac{\text{hop\_length}}{f_s}, \quad m = 0, 1, \ldots, M-1$$

  Katrs laika moments $t[m]$ atbilst loga centram laikā.

- **Kopējais laika skaits**: $M = \left\lfloor \frac{N - n_{\text{fft}}}{\text{hop\_length}} \right\rfloor + 1$, kur N ir kopējais paraugu skaits.

1.2 Nenoteiktības princips Furjē transformācijā

Vienas dimensijas daļiņas stāvokli apraksta viļņfunkcija $f \in L^2(\mathbb{R})$. Lai analizētu vienlaicīgu lokalizāciju fiziskajā un frekvenču telpā, vispirms definējam:

Definīcija 1 (Vidējā pozīcija un impulss)**.** Normētai funkcijai $\|f\|_{L^2} = 1$:

$$\langle x \rangle = \int_{-\infty}^{\infty} x |f(x)|^2 dx \quad \text{(vidējā pozīcija)} \tag{1.21}$$

$$\langle \xi \rangle = \int_{-\infty}^{\infty} \xi |\hat{f}(\xi)|^2 d\xi \quad \text{(vidējais impulss frekvenču telpā)} \tag{1.22}$$

kur $\hat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\xi x} dx$.

Definīcija 2 (Laika un frekvences dispersija)**.**

$$(\Delta x)^2 = \int_{-\infty}^{\infty} (x - \langle x \rangle)^2 |f(x)|^2 dx \quad \text{(laika dispersija)} \tag{1.23}$$

$$(\Delta \xi)^2 = \int_{-\infty}^{\infty} (\xi - \langle \xi \rangle)^2 |\hat{f}(\xi)|^2 d\xi \quad \text{(frekvences dispersija)} \tag{1.24}$$

1.2.1 Heizenberga-Pauli-Veila nevienādība

Fundamentālo ierobežojumu vienlaicīgai lokalizācijai apraksta:

Teorēma 1 (Heizenberga-Pauli-Veila nevienādība)**.** *Jebkurai funkcijai* $f \in \mathcal{S}(\mathbb{R})$ *ar* $\|f\|_{L^2} = 1$ *ir spēkā:*

$$(\Delta x)(\Delta \xi) \geq \frac{1}{2} \tag{1.25}$$

Vienādība ir spēkā tad un tikai tad, ja f *ir modulēta Gausa funkcija:*

$$f(x) = Ce^{-\alpha(x-x_0)^2} e^{i\xi_0 x} \tag{1.26}$$

kur $\alpha > 0$*,* $C \in \mathbb{C}$*, un* $x_0, \xi_0 \in \mathbb{R}$*, un* $\mathcal{S}$ *ir Schwartz space*

Pierādījums. Izmantojot kanonisko komutācijas sakarību starp pozīcijas un impulsa operatoriem [3]:

$$[x, -i\partial_x] = x(-i\partial_x) - (-i\partial_x)x = i,$$

un Košī-Švarca nevienādību [4], iegūstam:

$$(\Delta x)^2(\Delta \xi)^2 \geq \frac{1}{4} \left|\langle f, [x, -i\partial_x] f\rangle\right|^2.$$

Komutācijas sakarība dod:

$$\langle f, [x, -i\partial_x] f\rangle = i\langle f, f\rangle = i,$$

tātad:

$$(\Delta x)(\Delta \xi) \geq \frac{1}{2}.$$

Vienādība pastāv tad un tikai tad, ja operatoru novirzes $(x - \langle x\rangle)f$ un $(-i\partial_x - \langle \xi\rangle)f$ ir lineāri atkarīgas [5]:

$$(-i\partial_x - \langle \xi\rangle)f = i\alpha(x - \langle x\rangle)f.$$

Šī diferenciālvienādojuma atrisinājums ir Gausa funkcija:

$$f(x) = Ce^{-\alpha(x-x_0)^2} e^{i\xi_0 x}.$$

□

Šai teorēmai ir nozīmīgas sekas signālu apstrādē:

- Neviens ierobežotas enerģijas signāls nevar vienlaicīgi būt patvaļīgi lokalizēts laika un frekvences jomā.
- Laika-frekvences lokalizācijas kompromiss nosaka filtru dizainu un spektrogrammu izšķirtspēju.
- Tas ir pamatā Gabortransformācijām un viļņveidu analīzei.

1.3 Skaņas atpazīšana ar Mel Spektrogrammu

1.3.1 Mel-Skalas Filtri un Mel Spektrogrammas

Definīcija 3 (Mel filtri (trīsstūrveida))**.** Mel filtri ir trīsstūrveida joslas filtri, kas izvietoti pēc Mel skalas [6], kas ir perceptuāli lineāra frekvenču skalas tuvinājums cilvēka dzirdes uztverei. Katrs filtrs $H_m(k)$ ir definēts ar:

$$H_m(k) = \begin{cases} \frac{k - f_{m-1}}{f_m - f_{m-1}}, & f_{m-1} \leq k < f_m, \\ \frac{f_{m+1} - k}{f_{m+1} - f_m}, & f_m \leq k \leq f_{m+1}, \\ 0, & \text{citādi}, \end{cases}$$

kur f_m ir centrālās frekvences, pārveidotas no Mel skalas $m \leq M$ (parasti $M = 40$) izmantojot:

$$f_{\text{mel}} = 2595 \log_{10} \left(1 + \frac{f_{\text{Hz}}}{700} \right).$$

Motivācija

Mel filtri tiek izmantoti, lai:

- Samazinātu dimensiju: Kompresēt frekvenču asi no N STFT lodziņiem uz M Mel joslām ($M \ll N$) [7].
- Imitēt dzirdes uztveri: Cilvēka auss ir jutīgāka zemām frekvencēm izmaiņām($<$ 1 kHz), un Mel skalas filtri to modelē [8].
- Stabilizēt spektru: Trīsstūrveida svēršana samazina trošņu ietekmi.

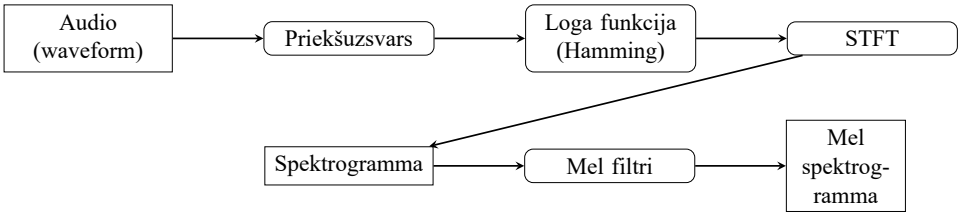
Pārveide no Spektrogrammas uz Mel Spektrogrammu

Dota STFT spektrogramma $S(t, k)$, Mel spektrogrammu $S_{\text{mel}}(t, m)$ iegūst:

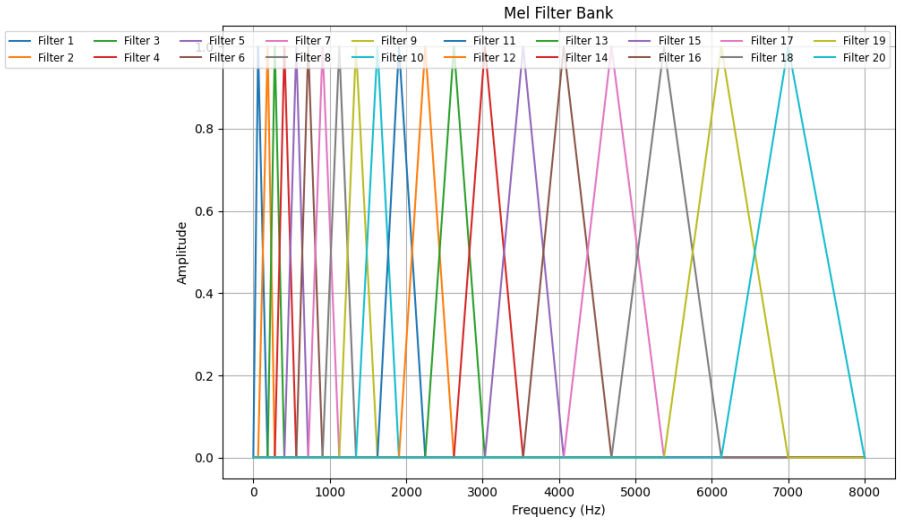
1. Log-enerģija: $\log |S(t, k)|^2$.
2. Mel filtru piemērošana:

$$S_{\text{mel}}(t, m) = \sum_{k=0}^{N-1} H_m(k) \cdot \log |S(t, k)|^2.$$

Signāla apstrādes ķēde/ceļš



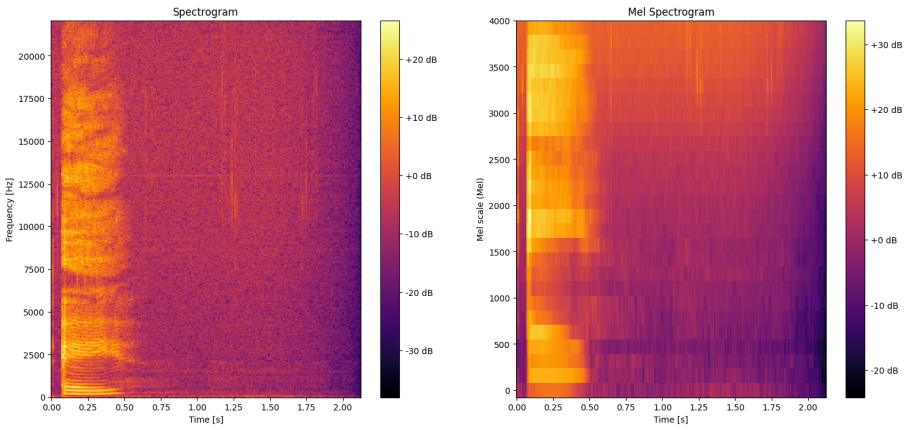
Att. 1.3: Signāla pārveides process no audio vilkņa uz Mel spektrogrammu.



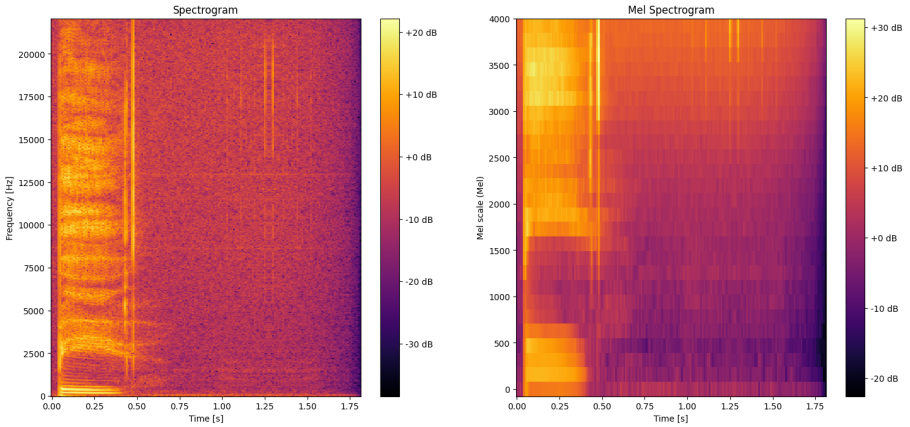
Att. 1.4: Mel filtri

1.3.2 Kopsavilkums

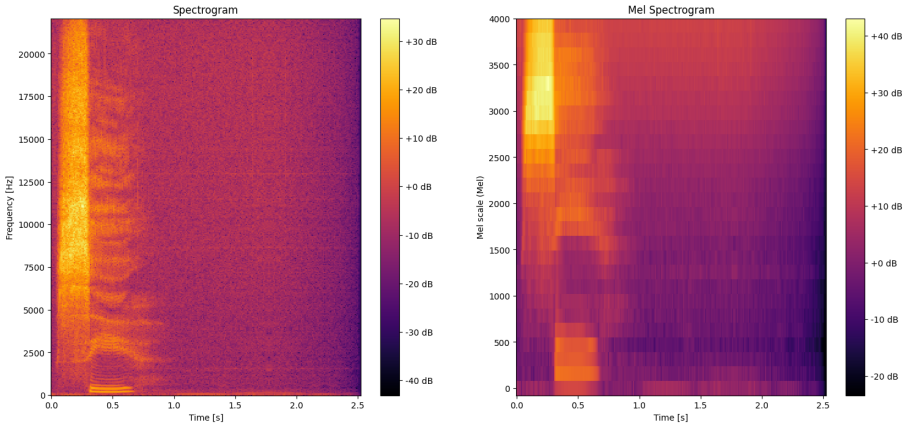
Mel spektrogrammas nodrošina kompaktu un cilvēka dzirdes sistemam atbilstošu skaņas reprezentāciju, kas piemērota tālākai analīzei (piem., emocionu klasifikācijai [9]). Tomēr, lai iegūtu vēl kompaktāku un interpretējamāku atveidi, izmanto **cepstrālu** (MFCC), kas tiks apspriests nākamajā nodaļā kopā ar slēpto Markova modeļu (HMM) pielietojumu.



Att. 1.5: Burta ’a’ izruna angļu valodā spektrogramma



Att. 1.6: Burta ’b’ izruna angļu valodā spektrogramma



Att. 1.7: Burta ’c’ izruna angļu valodā spektrogramma

1.4 MFCC un HMM

1.4.1 MFCC iezīmju izvilkšana

Pēc Mel spektrogrammas iegūšanas tiek veikti šādi soļi, lai iegūtu MFCC koeficientus:

1. **Logaritmiskā kompresija**: Mel spektrogrammas enerģijas tiek logaritmētas, lai samazinātu dinamiskā diapazona ietekmi:

$$E_{\text{mel}} = \log \left(\sum_{k=0}^{N_{\text{fft}}/2} |X[k]|^2 H_m[k] \right),$$

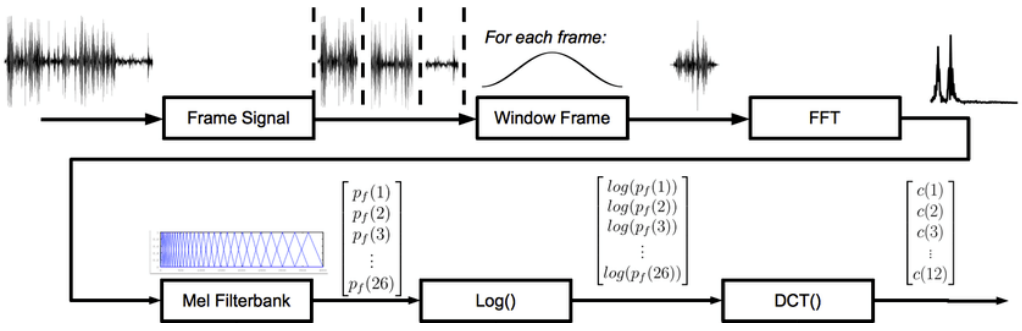
   kur $X[k]$ ir STFT spektrs, un $H_m[k]$ ir m-tā Mel filtra atbilde.

2. **Diskrētā kosinusa transformācija (DCT)**: Logaritmētās enerģijas tiek pārveidotas uz cepstrālo domēnu, izmantojot DCT:

$$c_i = \sum_{m=1}^{M} E_{\text{mel}}[m] \cos \left(\frac{\pi i (m - 0.5)}{M} \right), \quad 1 \leq i \leq L,$$

   kur M ir Mel filtru skaits, un L ir MFCC koeficientu skaits (parasti 13).

The Mel-frequency cepstrum coefficient (MFCC) calculation process is illustrated in Figure 1.8 [10].



Att. 1.8: Mel-frequency cepstrum coefficient (MFCC) calculation process.

1.4.2 Gausa slēptie Markova modeļi (HMM)

Gausa HMM tiek izmantoti, lai modelētu runas signālu secības. Katrs HMM sastāv no slēptiem stāvokļiem, kas apraksta signāla izmaiņas laikā.[11]

Definīcija 4 (Gausa HMM)**.** Gausa HMM ir definēts ar sekojošiem parametriem:

- N - slēpto stāvokļu skaits,
- M - novērojumu skaits (MFCC koeficientu dimensija),
- A - pāreju varbūtību matrica, kur a_{ij} apzīmē varbūtību pāriet no stāvokļa i uz stāvokli j,
- B - emisijas varbūtības, kas aprakstītas ar Gausa sadalījumiem:

  $$b_j(o_t) = \mathcal{N}(o_t; \mu_j, \Sigma_j),$$

  kur μ_j ir vidējā vērtība un Σ_j ir kovariācijas matrica stāvoklim j,
- π - sākotnējo stāvokļu varbūtības.

Apmācības process

HMM apmācība tiek veikta, izmantojot Baum-Welch algoritmu, kas maksimizē log-likelihood funkciju[12]:

$$\mathcal{L}(\lambda) = \log P(O|\lambda),$$

kur O ir novērojumu secība (MFCC koeficienti), un λ ir HMM parametri. Algoritms atjaunina parametrus iteratīvi:

$$\mu_j^{new} = \frac{\sum_{t=1}^{T} \gamma_t(j) o_t}{\sum_{t=1}^{T} \gamma_t(j)},$$

$$\Sigma_j^{new} = \frac{\sum_{t=1}^{T} \gamma_t(j)(o_t - \mu_j)(o_t - \mu_j)^T}{\sum_{t=1}^{T} \gamma_t(j)},$$

kur $\gamma_t(j)$ ir varbūtība atrasties stāvoklī j laikā t.

Novērtēšana

Pēc apmācības HMM var novērtēt jaunus novērojumus, izmantojot log-likelihood vērtību:

$$\text{score} = \log P(O|\lambda).$$

Šī vērtība tiek izmantota, lai noteiktu, kura HMM vislabāk atbilst ievades signālam.

1.4.3 Sistēmas darbība

Darbā izstrādāta sistēma izmanto MFCC iezīmes, lai apmācītu Gausa HMM katram ciparam no 1 līdz 9. Apmācības process balstās uz **2940 audio failiem**, kas ierakstīti no 6 dažādiem runātājiem, kur katrs runātājs katru ciparu izrunājis 49 reizes. Šie faili tika izmantoti, lai veidotu Gausa HMM, kas modelē runas temporālās un spektrālās īpašības. Testēšanas posmā tika izmantota atsevišķa **60 audio failu** kopa, kas arī ierakstīta no tiem pašiem 6 runātājiem, katrs cipars tika izrunāts 10 reizes. Šī testēšanas datu kopa ir savstarpēji nesakritīga ar apmācības datu kopu, taču abas kopas tika ierakstītas no tiem pašiem runātājiem un vienādos apstākļos.

Pēc apmācības modeļi tiek saglabāti un izmantoti, lai klasificētu jaunus audio signālus, aprēķinot to log-likelihood vērtības un izvēloties modeli ar augstāko vērtību.

2 Praktiskā analīze un pētījums

2.1 Testēšanas process

Testēšanas process sastāv no divām galvenajām daļām: modeļa apmācība un testēšana. Šajā sadaļā tiek aprakstīts Python kods un parametri, kas tika izmantoti šī procesa realizācijai. Tāpat tiek iekļauti koda fragmenti un to izvades, lai ilustrētu sistēmas darbību.

2.1.1 Modeļa apmācība

Apmācības process tika veikts, izmantojot `HMMTrainer` klasi, kas ir definēta kā daļa no sistēmas. Šī klase izmanto Gausa slēptos Markova modeļus (HMM), lai modelētu runas signālus. Galvenie parametri, kas tika izmantoti apmācībā, ir šādi:

- **Modela tips**: Gausa HMM (`model_type='GaussianHMM'`). Gausa HMM ir plaši izmantoti runas atpazīšanā, jo tie spēj efektīvi modelēt laikā mainīgus spektrālos raksturlielumus [11].
- **Stāvokļu skaits**: $N = 3$ (`n_components=3`). Šis parametrs tika izvēlēts, lai nodrošinātu pietiekami sarežģītu modeli, kas spēj atspoguļot runas signāla izmaiņas laikā, bet nav pārāk sarežģīts, lai izraisītu pārmācīšanos .
- **Kovariācijas tips**: Diagonāls (`cov_type='diag'`). Diagonālā kovariācija tika izvēlēta, lai samazinātu aprēķinu sarežģītību un izvairītos no pārmācīšanās [11].
- **Iterāciju skaits**: 2000 (`n_iter=2000`). Šis iterāciju skaits tika izvēlēts, lai nodrošinātu pietiekamu parametru konverģenci .

Apmācības dati sastāvēja no 2940 audio failiem, kas ierakstīti no 6 dažādiem runātājiem[13]. Katrs runātājs katru ciparu (no 1 līdz 9) izrunāja 49 reizes. Šie dati tika apstrādāti, lai iegūtu MFCC iezīmes, kas tika izmantotas HMM apmācībai.

Apmācības procesa izvade

Tālāk ir parādīta programmas izvade, kas tika iegūta apmācības procesa laikā:

```
--- Analysis for label '7' ---
Convergence information:
  - Iterations: 21
  - Converged: Yes
Transition matrix shape: (3, 3)
Transition matrix:
[[9.38664979e-01 1.23583326e-02 4.89766886e-02]
 [3.96124912e-02 9.58022076e-01 2.36543236e-03]
 [6.09908588e-02 6.22420456e-05 9.38946899e-01]]
Means shape: (3, 13)
Means:
[[ -8.01146672 -14.77155938   4.55285476  -4.68587769  -9.17583611
   -8.45346759  -5.92359407  -2.13979066  -8.11031235  -4.29281351
   -2.49723439  -9.31849351  -2.0864188 ]
 [ -2.3258759  -12.58260223  -5.27265863  -9.03861927 -28.5008378
  -35.49902068  15.21977476   8.80769216 -10.92823055  13.11976233
  -11.57984678  -4.138628    -3.74076534]
 [ -4.26822371  -1.12377142  -7.74427714  -8.60917027 -27.016354
   -9.66033725  -4.14066906  14.05429252 -12.31010058 -12.21453236
    7.32885024 -19.48533117  -0.0656589 ]]
Covariance shape: (3, 13, 13)
Covariance (first 5 rows):
[[[  8.13782936   0.           0.           0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.         183.58560667   0.           0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.           0.          75.07914088   0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.           0.           0.         102.43653052   0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.           0.           0.           0.         114.96016933
     0.           0.           0.           0.           0.
     0.           0.           0.        ]]
```

```
 [[  3.67256767   0.           0.           0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.          60.46643306   0.           0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.           0.          27.63644873   0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.           0.           0.          45.60002614   0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.           0.           0.           0.          65.02096524
    0.           0.           0.           0.           0.
    0.           0.           0.        ]]

 [[  5.61939611   0.           0.           0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.          41.59715363   0.           0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.           0.          79.29641628   0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.           0.           0.         119.36051096   0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
  [  0.           0.           0.           0.          55.56522111
    0.           0.           0.           0.           0.
    0.           0.           0.        ]]]


--- Analysis for label '8' ---
Convergence information:
  - Iterations: 32
  - Converged: Yes
Transition matrix shape: (3, 3)
Transition matrix:
[[9.17721935e-01 7.11271352e-30 8.22780646e-02]
```

```
 [4.64790230e-02 9.35522891e-01 1.79980863e-02]
 [1.16797172e-03 6.03047767e-02 9.38527252e-01]]
Means shape: (3, 13)
Means:
[[-1.29093174e+00 -1.48852155e+01  5.53684018e+00 -1.20142769e+01
  -4.26009573e+01 -5.12373455e+00  3.91036071e+00 -2.62299651e+00
  -7.63547344e-01  5.22704357e+00 -1.75455268e+01  1.62079347e+00
  -4.10438683e+00]
 [-8.24925047e+00 -2.13046151e+01  4.03621871e-02 -7.76673625e+00
  -1.39406871e+01 -5.86222426e+00 -2.96466573e+00 -4.30818822e+00
  -9.45703646e-01  2.01184806e+00 -9.66045881e-01 -5.38377849e+00
   3.49079559e+00]
 [-4.77812125e+00 -8.31548005e+00  1.90595200e+01 -1.09803990e+01
  -3.83526953e+01 -6.08506784e+00 -1.99703527e+01 -8.04551640e+00
   7.20888751e+00 -4.97486623e-01 -7.87702634e-01 -1.24848244e-01
  -7.89075291e+00]]
Covariance shape: (3, 13, 13)
Covariance (first 5 rows):
[[[  2.87526385   0.           0.           0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.          42.57553934   0.           0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.           0.         132.99959838   0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.           0.           0.          80.61575686   0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.           0.           0.           0.         112.06672739
     0.           0.           0.           0.           0.
     0.           0.           0.        ]]

 [[  6.98165865   0.           0.           0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.          57.27334574   0.           0.           0.
     0.           0.           0.           0.           0.
     0.           0.           0.        ]
  [  0.           0.          72.56807914   0.           0.
```

```
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
 [  0.           0.           0.          83.60035903   0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
 [  0.           0.           0.           0.         153.85033393
    0.           0.           0.           0.           0.
    0.           0.           0.        ]]

[[  7.44281054   0.           0.           0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
 [  0.          31.95884213   0.           0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
 [  0.           0.          54.04914599   0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
 [  0.           0.           0.          90.40884788   0.
    0.           0.           0.           0.           0.
    0.           0.           0.        ]
 [  0.           0.           0.           0.         125.42867699
    0.           0.           0.           0.           0.
    0.           0.           0.        ]]]
```

Slēpto Markova modeļu (HMM) apmācība uz etiķetētiem raksturlielumu datiem 2.1.2.2
Process ietver raksturlielumu (MFCC) iegūšanu no apmācības datiem, to grupēšanu pēc etiķetēm un HMM apmācību katrai etiķetei, izmantojot norādītos parametrus. Apstrādes progresu uzrāda progresa josla

```
117     def process_training_directory(self):
118         mfcc_features = {}
119         for filename in glob("{0}/*.wav".format(self.training_directory)):
120 
121             # Read the input file
122             audio, sampling_freq = sf.read(filename)
123             label = self.get_training_label(filename)
124             features = mfcc(audio, sampling_freq, nfft=self.nfft)
125 
126             # Group by label
127             if label not in mfcc_features:
128                 mfcc_features[label] =([],[])
129             mfcc_features[label][0].append(features)
130             mfcc_features[label][1].append(len(features))
131 
132         for label in mfcc_features:
133             features_list, lengths = mfcc_features[label]
134             mfcc_features[label] = (np.concatenate(features_list, axis=0), lengths)
135         return mfcc_features
```

Att. 2.1: Koda fragments apmācības direktorijas apstrādei .

```
89      def train_model(self):
90          training_files = []
91          print("Processing training data...")
92          grouped_features = self.process_training_directory()
93          print("Training data processed.")
94 
95          # Train HMM for each MFCC and add to training set
96          for label,(X,lengths) in tqdm(grouped_features.items()):
97              hmm_trainer = HMMTrainer(model_name=self.model_type,
98                  n_components=self.n_components,
99                  cov_type=self.cov_type,
100                 n_iter=self.n_iter)
101             hmm_trainer.train(X,lengths)
102             training_files.append({'label': label, 'hmm_trainer': hmm_trainer})
103 
104         return training_files
```

Att. 2.2: Koda fragments datu apmacībai

2.1.2 Testēšanas process

Testēšanas process tika veikts, izmantojot atsevišķu testēšanas datu kopu, kas sastāvēja no 60 audio failiem. Šie faili tika ierakstīti no tiem pašiem 6 runātājiem, katrs cipars tika izrunāts 10 reizes. Testēšanas process sastāv no šādiem soļiem:

1. **Audio failu ielasīšana**: Katrs testēšanas audio fails tika ielasīts, izmantojot `soundfile` bibliotēku.
2. **MFCC iezīmju izvilkšana**: Katram audio failam tika aprēķinātas MFCC iezīmes, izmantojot `python_speech_features` bibliotēku. Parametri:
   - **FFT punktu skaits**: 1203 (`nfft=1203`).
   - **Mel filtru skaits**: 20.
   - **MFCC koeficientu skaits**: 13.
3. **Modeļa novērtēšana**: Katram testēšanas failam tika aprēķināta log-likelihood vērtība visiem apmācītajiem HMM. Tika izvēlēts modelis ar augstāko log-likelihood vērtību, lai noteiktu atpazīto ciparu.

```
Processing training data...
label: 0
mfcc features shape: (58, 13)
mfcc features (first 5 rows): [[ -6.167823    10.16323209  21.6439572   -4.34020536 -12.99802226
  -31.69034521  -6.31665194 -15.52176628 -17.55438551   3.45207941
   -4.303359   -12.09215598  -3.95266519]
 [ -5.76623668   8.43086204  18.08325232  -3.61721704  -9.95103974
  -29.80239752  -3.1347134  -20.18776348 -25.90461774  13.53624698
   -8.40207909  -6.08568419   1.32857729]
 [ -5.72489936   4.93374078  18.03799334  -4.70194739 -10.64036543
  -32.62734324  -6.85643201 -20.93727176 -32.09308802   6.75021803
  -10.55326875   1.77545775   2.23348344]
 [ -5.77290814   1.15132123  13.65508211 -10.17577039  -8.65562229
  -34.64379456 -10.60496465 -12.89987134 -14.90426898  10.41647854
  -12.44816569  -2.06425507   0.8209956 ]
 [ -5.70984929   0.5989776   15.31496373 -15.25317611  -6.69469758
  -33.82556144 -10.17322919 -10.38347202 -10.6068623    8.73973869
  -11.6732231    1.93081839  11.18872818]]
```

Att. 2.3: Koda fragments ar MFCC matricu

2.1.3 Modeļa analīze

```
Accuracy: 91.66666666666666%
Model parameters are: nfft=1203, model_type=GaussianHMM,
    ↪ n_components=3, cov_type=diag, n_iter=2000

Accuracy: 96.66666666666667%
Model parameters are: nfft=1203, model_type=GaussianHMM,
    ↪ n_components=4, cov_type=diag, n_iter=2000

Accuracy: 95.0%
Model parameters are: nfft=1203, model_type=GaussianHMM,
    ↪ n_components=5, cov_type=diag, n_iter=2000

Accuracy: 96.66666666666667%
Model parameters are: nfft=1203, model_type=GaussianHMM,
    ↪ n_components=6, cov_type=diag, n_iter=2000

Accuracy: 98.33333333333333%
Model parameters are: nfft=1203, model_type=GaussianHMM,
    ↪ n_components=10, cov_type=diag, n_iter=2000

Accuracy: 98.33333333333333%
Model parameters are: nfft=1203, model_type=GaussianHMM,
    ↪ n_components=15, cov_type=diag, n_iter=2000

Accuracy: 91.66666666666666%
Model parameters are: nfft=800, model_type=GaussianHMM,
    ↪ n_components=20, cov_type=diag, n_iter=2000

Accuracy: 91.66666666666666%
Model parameters are: nfft=800, model_type=GaussianHMM,
    ↪ n_components=3, cov_type=diag, n_iter=2000

Accuracy: 93.33333333333333%
Model parameters are: nfft=800, model_type=GaussianHMM,
    ↪ n_components=3, cov_type=diag, n_iter=1000

Accuracy: 98.33333333333333%
Model parameters are: nfft=800, model_type=GaussianHMM,
    ↪ n_components=10, cov_type=diag, n_iter=1000

Accuracy: 96.66666666666667%
Model parameters are: nfft=800, model_type=GaussianHMM,
    ↪ n_components=30, cov_type=diag, n_iter=1000
```

Rezultāti parāda, ka stāvokļu skaita palielināšana var uzlabot precizitāti, bet šī ietekme **nav lineāra** un **nav viennozīmīga**. Galvenais secinājums ir, ka **precizitāte ir augsta** visos gadījumos, un nav iespējams izdarīt skaidrus secinājumus par

parametru ietekmi uz precizitāti. **Šis rezultāts ir pretrunā teorijai**, kas paredz, ka lielāks stāvokļu skaits (`n_components`) pasliktinā modeļa darbību, jo tas sāk modelēt troksni vai pārmācīties.

2.1.4 Izmantotās bibliotēkas un kods

Šajā darbā tika izmantotas divas Python bibliotēkas: `python_speech_features` MFCC iezīmju izvilkšanai un `hmmlearn` slēpto Markova modeļu (HMM) apmācībai. [14, 15]. Darbā izstrādātais kods :[16].

Secinājumi

2.1.5 Rezultātu ticamības analīze

Eksperimentālie rezultāti parāda augstu precizitāti ($> 91\%$), tomēr rodas šaubas par to interpretāciju:

- **Parametru ietekmes nenoteiktība**: Palielinot slēpto stāvokļu skaitu (N), netika novērots sistemātisks precizitātes kritums, kas pretrunā teorētiskajai prognozei. Teorētiski, $N \gg$ patiesais stāvokļu skaits noved pie pārmācīšanās, taču eksperimenti to neapstiprina.
- **Iespējamie izskaidrojumi**: Datu viendabība: Apmācības dati iegūti no **6 runātājiem** identiskos ierakstu apstākļos, kas samazina variāciju un nepieciešamību pēc sarežģītiem modeļiem.

2.1.6 Runas atpazīšanas sistēmas arhitektūras kopsavilkums

Piedāvātā sistēma balstās uz trīs pamatkomponentiem:

- **Mel spektrogrammas**: Pārveido signālu laika-frekvences telpā, imitējot cilvēka dzirdes uztveri:

$$f_{\text{mel}}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

- **MFCC raksturlielumi**: Izolē spektrālās enerģijas dinamiku, izmantojot diskrēto kosinusa transformāciju:

$$c_n = \sum_{k=1}^{K} \log E_k \cdot \cos\left(\frac{n(k-0.5)\pi}{K}\right), \quad 1 \leq n \leq L$$

- **Slēptie Markova modeļi**: Modelē stāvokļu pārejas ar matricu $A = [a_{ij}]$ un emisijas varbūtības $B = [b_j(o_t)]$.

2.1.7 Turpmākie pētījumu virzieni

Lai uzlabotu sistēmas robustumu un interpretējamību, ieteicams:

- **Alternatīvu modeļu izmantošana**:
  - Dziļie neironu tīkli (DNN) ar pašuzmanības mehānismiem
  - Transformatori spektrālo raksturlielumu analīzei un modelēšanai
- **Turpmāka teorētiskā izpēte**:
- **Datu paplašināšana (piemēram, ar Gausa troksni)**: Ģeneratīvi mākslīgie dati ($\tilde{X} = X + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$), lai samazinātu pārmācīšanos.

Beigu secinājums: Šajā darbā tika analizētas dažādas signālu apstrādes metodes un izstrādāts efektīvs modelis runas atpazīšanai. Tomēr turpmāki pētījumi ir nepieciešami, lai uzlabotu modeļa precizitāti un piemērojamību sarežģītākos apstākļos.

Pateicības

Pirmkārt, es vēlos pateikties **Liang Wenfeng** par to, ka viņš man iedvesmoja un palīdzēja izpildīt šo kursa darbu. Lai arī viņš droši vien nezina, ka es pastāvēju, viņa idejas un darbi man deva spēku turpināt, pat kad viss šķita bezcerīgi. Arī īpaša pateicība manai **tējkannai**, kas nepārtraukti vārīja ūdeni manām zaļās tējas porcijām, un manam gultas spilvenam, kas pacietīgi gaidīja mani katru rītu. Paldies!

Izmantotā literatūra un avoti

[1] ScienceDirect. Hamming window, 2023. Accessed: 2025-01-22.

[2] Brian McFee. Aliasing, 2023. Accessed: 2023-01-22.

[3] David J. Griffiths. *Introduction to Quantum Mechanics*. Cambridge University Press, Cambridge, 2018. Kanoniskās komutācijas sakarības.

[4] Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics*, volume 1. Academic Press, New York, 1980. Košī-Švarca nevienādība operatoriem.

[5] Brian C. Hall. *Quantum Theory for Mathematicians*. Springer, New York, 2013. Vienādības nosacījums nenoteiktības principam.

[6] S.S. Stevens and J. Volkmann. The relation of pitch to frequency. *American Journal of Psychology*, 1940.

[7] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1980.

[8] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[9] B. Schuller. *Intelligent Audio Analysis*. Springer, 2013.

[10] ResearchGate. Figure 4. mel frequency cepstrum coefficient (mfcc) calculation. `https://www.researchgate.net/publication/277553387/figure/fig4/AS:614320232734758@1523476765884/Mel-frequency-cepstrum-coefficient-MFCC-calculation.png`, 2025. Accessed: 2025-01-23.

[11] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989.

[12] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 1970.

[13] Zvonimir Jakobovski. Free spoken digit dataset (fsdd). `https://github.com/Jakobovski/free-spoken-digit-dataset`, 2025. Accessed: 2023-10-15.

[14] James Lyons. python_speech_features: A python library for audio feature extraction. `https://github.com/jameslyons/python_speech_features`, 2025. Accessed: 2023-10-15.

[15] hmmlearn Developers. hmmlearn: A python library for hidden markov models. `https://hmmlearn.readthedocs.io/`, 2025. Accessed: 2023-10-15.

[16] Kirills Bobkovs. Speech recognition code. `https://github.com/Benslic/kursa_darbs`, 2025. GitHub repository.

Bakalaura darbs “Slēpto Markova modeļu sistēmas izstrāde runāto ciparu atpazīšanai, izmantojot MFCC pazīmes.” izstrādāts LU Fizikas un Matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Kirills Bobkovs

_____ _____
(paraksts) (datums)

Rekomendēju darbu aizstāvēšanai.
Vadītājs: prof. Dr.math. Uldis Strautiņš

_____ _____
(paraksts) (datums)

Recenzents:

_____ _____
(paraksts) (datums)

Darbs iesniegts Matemātikas nodaļā _____
(datums)

_____
(darbu pieņēma)

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums_____
(datums)

Komisijas sekretārs/-e: _____ _____
(Vārds, Uzvārds) (paraksts)