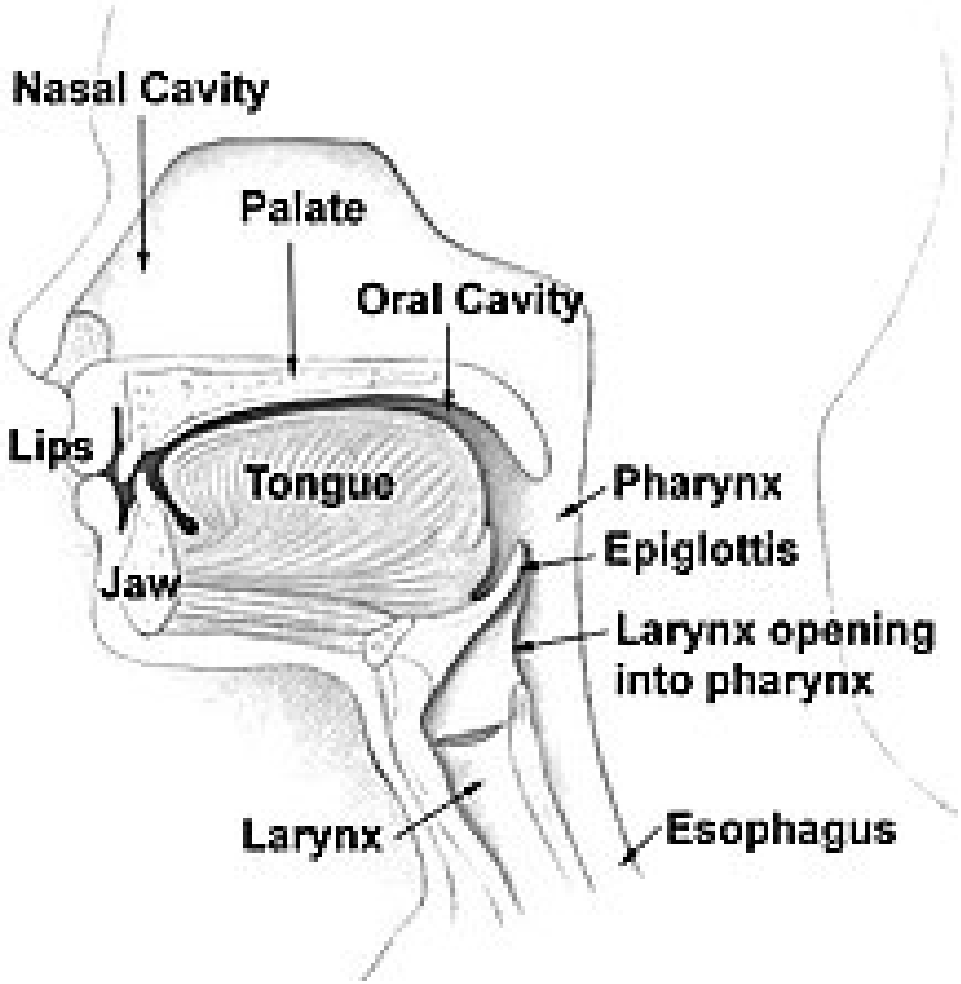
Identification Systems

Lecture 14

Piotr Syga

2021-06-15

Voice aparatus schema



Idea

- together with facial recognition the most common 'biometric' method of people identifying each other
- we utilize the tone, pace and frequency of speech
- we use a mapping to a domain mimicking human hearing
- auxiliary parameters include fixed passphrase, the language or the volume
- main issues to consider: noise, reflected voice, multiple speakers, resistance to presentation attacks

Speaker recognition vs speech recognition

Speaker recognition – set of methods that aims to identify the person that speaks, main tools include frequency domain analysis and analysis of the components. May utilize the identification of the words that are spoken as an additional feature.

Speech recognition – set of methods that aims to recognize the words that are spoken, often included in speech-to-text tools; mainly utilizes pattern recognition and dictionary methods (syllable and word recognition). May utilize personalized speaking patterns as an additional feature.

Milestones

- speech recognition milestones:
  - 1950s spoken digits recognition
  - 1960s IBM Shoebox – recognition of 16 words
  - 1970s Harpy recognizes over 1000 words
  - 1980s HMM used for phonemes prediction
- speaker recognition:
  - 1960s Fant's X-Ray photos allow observing unique patterns
  - 1970s pattern analysis with windowed Fourier Transform, first prototypes (Texas Instruments) and first patents (Fuji Xerox)
  - 1980s Voice Activity Detection and separating noise from speech
  - 1990s GMM used for identifying 'best match' speaker
  - 2000s deriving new features and testing various classifiers
  - 2010s ML approach

Types of recognition

fixed text – a fixed text (passphrase) is used during the enrollment and authentication procedures, the recognition of the passphrase may be a step in user authentication

dictionary – we have a set of fixed texts that may be used to synthesize a passphrase, the spoken words may be selected either by the authenticating system (as a challenge) or by the user

free speech – identifying a person based on a any spoken text or conversation, may also have a (hidden) passphrase incorporated into the sentences

Methods of acquisition

- online vs offline verification
- omnidirectional vs directional microphone

Sample representation

A sample (or a single frame of the sample) may be represented in various domains:

- frequency domain – as a spectrum or logarithm of the spectrum of the voice
- time domain – as a set (or continuous range) of values or signal's energy
- time-frequency domain – as a 3D map, were the values are represented for each time-frequency pair, this includes spectrograms and voiceprints

Preprocessing

- Noise Reduction: spectral analysis (FFT or cosine transform) and multithresholding (e.g., Otsu), filtering (e.g., FIR)
- Voice Activity Detection: splitting into equilength frames, filling the discontinuities obtaining signal parameters for frame classification, dividing frames into noisy ones and voice ones, analyzing only voice frames
- Separating frames of various speakers

Preprocessing

FIR

- a filtering signal that settles to 0 in finite time
- $S(t) = \sum_{j=0}^{N} a_j s(t-j)$
- order of the filter determines the number of non-zero a_js; 1st and 2nd order filters are commonly used (i.e., $\forall_{j>1} a_j = 0$ or $\forall_{j>2} a_j = 0$)
- FIRs are stable and require no feedback
- FIRs may be used in windows or per sample

Preprocessing

VAD

- splitting signal into equilength frames, $\hat{S}_l(t) = S(t; t + \alpha l)$, the frames may be disjoint or overlap
- filling the discontinuities, using windowed filtering $w(t) = \alpha - (1 - \alpha)\cos\frac{2t\pi}{C}$, $\hat{S}_{l;w}(t) = \hat{S}_l(t)w(t)$
- detecting the voice activity by measuring the energy $E = \sum(\hat{S}_{l;w}(t))^2$ and applying thresholds

Preprocessing

Long-Term Spectral Divergence

- we calculate 'the probability of voice event occurring'
- we divide the signal $s(t)$ on overlapping frames
- we assume that the signal consists of voice v and noise η, hence, for a frequency ω, we have $s(t,\omega) = v(t,\omega) + \eta(t,\omega)$
- we calculate $\alpha(t,\omega) = \text{argmax}_\tau \left| s(t+\tau,\omega) \right|$, for $\tau \in [-T, T]$ and the size (in samples) of the frame T that has to be carefully selected
- long-term spectral divergence is calculated as $\gamma(t) = \frac{1}{|\omega|_k} \sum_\omega \left(\frac{\alpha(t,\omega)}{\alpha_n(\omega)} \right)^2$, where $\alpha_n(\omega)$ is a spectral noise estimation, and $|\cdot|_k$ denotes the number of samples

Preprocessing

LTSD cont.

- the voice is classified based on a threshold

$$\beta_T(t) = \begin{cases} \beta_0, \gamma(t) \leq \gamma_0 \\ \beta_1, \gamma(t) \geq \gamma_1 \\ \frac{\beta_0 - \beta_1}{\gamma_0 - \gamma_1}\gamma(t) + \gamma_0(1 - \frac{\beta_0 - \beta_1}{1 - \gamma_1 - \gamma_0}), \text{oth.} \end{cases},$$

  with γ_0 and γ_1 are the expected PSNR for voice and noise frames, respectively and β_0 and β_1 are constant, depending on the sigmoidal activation function

- the first threshold frames are the voice frames, the second are the noise frames, the last are the frames with distorted voice

Mel scale

- it is a nonlinear mapping of the frequencies to mimic the human hearing
- it is based on subjective, human-hearing- not wavelength-related, distance between sounds
- typical (but not unique) conversion:
  $F_m = 1127 \ln(1 + \frac{f}{700}) \approx 1626 \log_2(1 + \frac{f}{700})$

Sample features

Time-domain extraction

- we assume that the noise $\eta(t)$ has a constant variance
- we build an auto-regressive model of the order O as $S(t) = \sum_{j=1}^{O} a_j s(t-j) + \eta$
- the signal is not windowed
- we estimate a_js using MSE and for a given threshold A, set $a_j = 0$ if they are below the threshold A
- optionally, we perform polynomial approximation from the adjusted a_js

Sample features

Frequency-domain extraction

- we perform spectral transform (FFT or cosine) using Hamming or triangular windows
- we perform quantization (similar to JPEG compression)
- we fix the order k and select k most significant FFT/cosine coefficients

Sample features

Mel-Frequency Cepstral Coefficients

- we extract the signal's features by utilizing power spectrum of the voice samples
- the voice sample is windowed (e.g., using Hamming window)
- for each windowed sample, FFT is computed
- the power of the spectrum is mapped into mel scale (using triangular or cosine windows) and logarithm is applied to the mapped signal to obtain the energy
- cosine transform (usually type IV) of the energy is calculated
- we take the coefficients of the amplitudes of the resulting spectrum

Sample features

Linear Predictive Coding

- can be used in a similar manner to MFCC, as its alternative
- we assume that in short samples, n-th signal is a linear combination p previous ones
- $S(t) = \sum_{j=1}^{p} a_j s(t-j)$
- to estimate a_js we invoke Levinson-Durbin algorithm, for efficient minimization of $\mathrm{MSE}(S(t)) = E[(S(t) - s(t))^2]$.
- the LPC utilizes windowing similar to MFCC, in order to divide the signal into frames and some k LPC coefficients are taken as features

Idea

Similar to other biometric modalities, after feature extraction we require a classification algorithm that matches the obtained sample to the closest enrolled identity (or verifies the closeness of the sample to given identity). Often used classifiers:

- k-Nearest Neighbour
- representing the feature vector in d-dimensional space and performing k-Means clustrization during the enrollment, the verified sample is mapped as a point and we assign the identity based on the cluster the sample ends in
- SVM
- Gaussian mixture model
- Boltzmann machine
- neural networks

GMM

- Gaussian Mixture Model is a sum of k normal distributions.
- It is an unsupervised learning model that assigns the data (e.g., identified sample) to one of the clusters, based on 'best matched normal distribution'.
- We assume that the probability of feature occurrence for some vector x in the model equals $p(x|w_i, \mu_i, \Sigma_i) = \sum_i^k w_i \mathcal{N}(x|\mu_i, \Sigma_i)$, for the normalized weights w_i, experimental mean μ_i, covariance matrix Σ_i and $\mathcal{N}(x|\mu_i, \Sigma_i) = ((2\pi)^{\frac{d}{2}} \sqrt{|\Sigma_i|})^{-1} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right)$.

GMM

- We perform Expectation–Maximization, namely
  - for each vector we calculate the probability distribution $p(x|w_i, \mu_i, \Sigma_i)$
  - we establish a (hidden) auxiliary variable $Z = [z_{ji}]$, for ith feature vector and jth normal distribution
  - we modify the parameters, so that the utility is maximized, namely for $\theta = \{w, \mu, \Sigma\}$ and its estimate θ', we obtain $Q(\theta, \theta') = E[\log p(X, Z)|\theta]$
  - using Lagrange multipliers we get $J(\theta, \theta') = Q(\theta, \theta') - \lambda(\sum_{i=1}^{k} w_i - 1)$
  - $\Pr[i|x_j] = \frac{w_i \mathcal{N}(x_j|\mu_i', \Sigma_i')}{\sum_{i=1}^{k} w_k \mathcal{N}(x_j|\mu_i', \Sigma_i')}$
  - hence we derive $n_i = \sum_j^N \Pr[i|x_j]$, $\mu_i = \frac{1}{n_i} \sum_{j=1}^{N} \Pr[i|x_j] x_j$, $w_i = \frac{n_i}{N}$ and $\Sigma_i = (\frac{1}{n_i} \sum_j^N \Pr[i|x_j] \mathrm{diag}(x_j, x_j^T)) - \mathrm{diag}(\mu_i', \mu_i'^T)$
- After the training finishes we can estimate the probability of the match of each of the samples.

Sample identification method

Training:

- We start with noise filtering and VAD.
- We derive MFCC, optionally we may derive LPC coefficients for simulating the pure voice.
- We may, optionally, add Universal Background Model classifier for simulating additional speakers.
- We organize the enrolled database using R-Trees.

User Identification:

- We denoise the sample and detect the speech.
- We derive the coefficients as during entrollemnt
- We send the samples to the classifier – GMM or Continuous Restricted Boltzmann Machine.
- To speed up the comparison we hierarchically search the database w.r.t. on the most significant coefficients.

Speaker recognition undernoisy and unconstrained conditions

J.S. Chung et al: *VoxCeleb2: Deep Speaker Recognition*

- development and comparison of several CNN models and training stategies
- the methods evaluated on VoxCeleb2 dataset – over million samples and 6000 speakers
- the authors obtain EER as low as 3.95% for ResNet-50 and under 6% for VGG-M
- 3 Titan X GPUs are used for the training

Speaker recognition undernoisy and unconstrained conditions

VGGVox

- from each frame features are extracted and pooled to obtain utterance-level speaker embeddings
- the model is trained on VoxCeleb2 set and tested on VoxCeleb1
- VGG-M is used as a base with one fc6 layer by two fully-connected 9x1 layers and 1xn average-pooling
- ResNet architectures have average pool layer added at the end, the rest of the architecture remains unchanged from the oryginal
- for training contrastive loss is used during training, by minimizing the distance between positive embeddings and maximizing the distance between negative ones, small α margin is used (pretraining with softmax and second stage fine tuning is performed to avoid local optima)
- time augmentation is tested by adding apool6 layer and temporal cropping of 3s segments
- training lasts 30epochs (with early stopping), SGD with batchsize of 64 is used for optimization, and learning rate logarithmically decreases from 10^{-2} to 10^{-8}.

Hidden Voice Attack

H. Abdullah et al.: *Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems*

- hidden attack is an attack that is obfuscated for human hearing by the artificial noise, yet enables an authentication or identification trial in a speaker recognition system
- the authors relax the constraint of requiring white-box knowledge of the identification system to perform a hidden attack
- the model-agnostic attack utilizes the common features of various models
- 4 classes of perturbations are derived to enable such attack
- 12 machine learning models (including Google's, Bing's, Mozilla's, IBM's and Azure) are tested against the attack
- all testes systems were vulnerable to the attack
- the attacks were less successful in noisy environments due to key-phrase distortion
- the research is based on the assumption of the limited number of features (most of which is removed in the preprocessing) that can be derived from human speech

Hidden Voice Attack

Perturbation Types

- Time Domain Inversion – utilizes the property of FFT that maps many signals to the same spectrum, they exploit it by modifying the audio in the time domain while preserving its spectrum, by inverting the windowed signal, due to the reversal the signal becomes unintelligible to humans
- Random Phase Generation – utilizes the fact that two different spectra obtained from FFT may have the same magnitude, so a second signal is created such the spectrum has parameters $c \neq a$ and $d \neq b$, but the magnitude $Y = \sqrt{a^2 + b^2 i} = \sqrt{c^2 + d^2 i}$.
- High Frequency Addition – utilizes the fact that during preprocessing higher frequencies are removed, by adding mixture of high frequency noise, the sample becomes unintelligible and with high frequencies may be painful to listen to
- Time Scaling – the samples is sped up, so it becomes shorter, unintelligible to humans, yet it retains the same spectrum

Hidden Voice Attack

Generation of attack audio

- The parameters are derived from within the set bounds. The bounds are derived so that the resulting samples remain unintelligible to humans.
- The legitimate audio and the matrix of the parameters are moved to perturbation engine, which derives a series of attack audio samples.
- The samples are created using a single perturbation type and composition of the perturbations.
- The samples are fed to the speech or speaker recognition system.
- The attack is declared successful if at least one sample is properly recognized.
- In the case of failure, the process is repeated with a new parameter set.

Further reading

- J. Ramirez, J. M. Gorriz, J. C. Segura. Voice activity detection. fundamentals and speech recognition system robustness.
- M. Ravanelli, Y. Bengio. Speaker Recognition from Raw Waveform with SincNet
- D. Snyder et al. Speaker Recognition for Multi-speaker Conversations Using X-vectors
- J.S. Chung et al. In defence of metric learning for speaker recognition
- C.S. Greenberg et al. Two decades of speaker recognition evaluation at the national institute of standards and technology
- Vox Celeb2 database
- sample githubs: *Unbounded Interleaved-State Recurrent Neural Network* for data clastrization, *SincNet* for processing audio samples, *SpeechBrain* general speech tool for speaker recognition, speech recognition and voice enhancement