Speaker verification

Steve Renals

Automatic Speech Recognition – ASR Lecture 17
18 March 2019

Speaker recognition

- Speaker identification – determine which of the set of enrolled speakers a test speaker matches
- Speaker verification – determine if a test speaker matches a *specific speaker*
- Speaker diarization – "who spoke when" segment and label a continuous recording by speaker

Speaker recognition

- Speaker identification – determine which of the set of enrolled speakers a test speaker matches
- **Speaker verification** – determine if a test speaker matches a *specific speaker*
- **Speaker diarization** (next lecture) – "who spoke when" segment and label a continuous recording by speaker

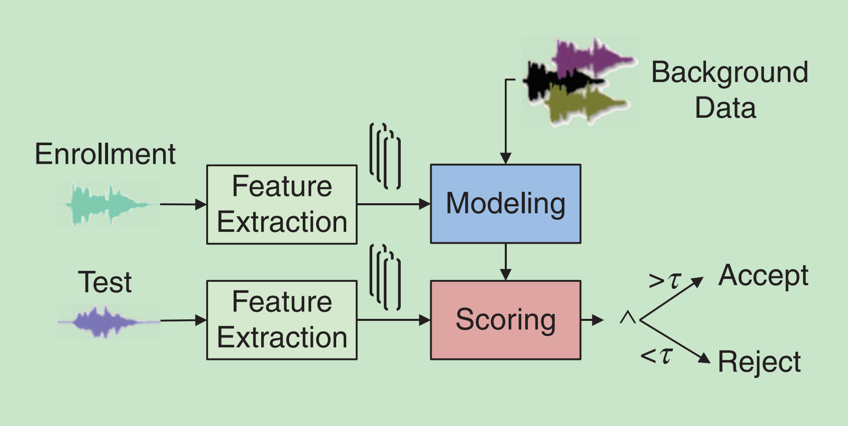
Speaker recognition

- Speaker identification – determine which of the set of enrolled speakers a test speaker matches
- **Speaker verification** – determine if a test speaker matches a *specific speaker*
- **Speaker diarization** (next lecture) – "who spoke when" segment and label a continuous recording by speaker

- Text dependent (vs text independent) – for speaker identification and verification, is the test speaker speaking a pre-defined utterance?
  - text-dependent – e.g. spoken password
  - text-independent – e.g. recognise a speaker from a law-enforcement recording
- Closed set (vs open set) – is there a fixed set of speakers?

Speaker verification

Overview of a speaker verification system



Source: Hansen and Hasan, 2015

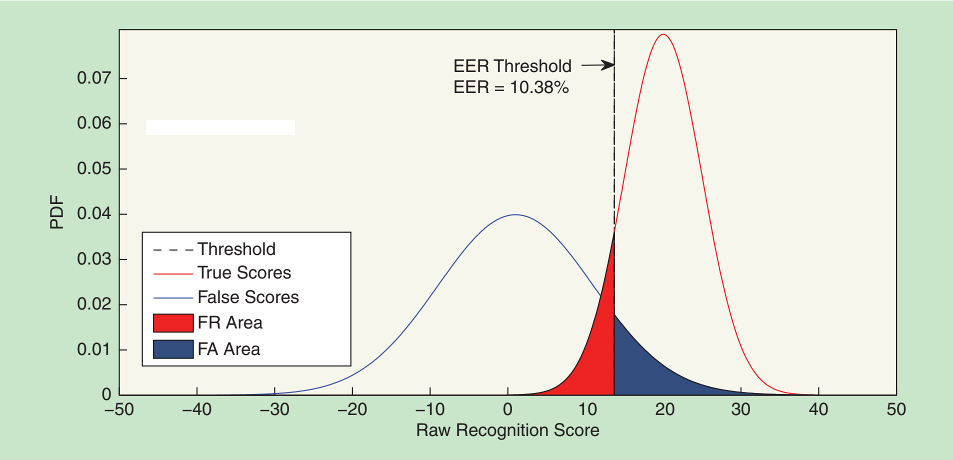
Evaluating speaker verification

- Two types of error
  - False acceptance – grant access to an imposter: **False Acceptance Rate (FAR)**
  - False reject – refuse access to a genuine speaker: **False Rejection Rate (FRR)**

$$\begin{aligned}
\text{FAR} &= \text{False Alarm Probability} \\
&= \frac{\text{Number of imposters accepted}}{\text{Number of imposter attempts}} \\
\text{FRR} &= \text{Miss Probability} \\
&= \frac{\text{Number of legitimate speakers rejected}}{\text{Number of legitimate attempts}}
\end{aligned}$$

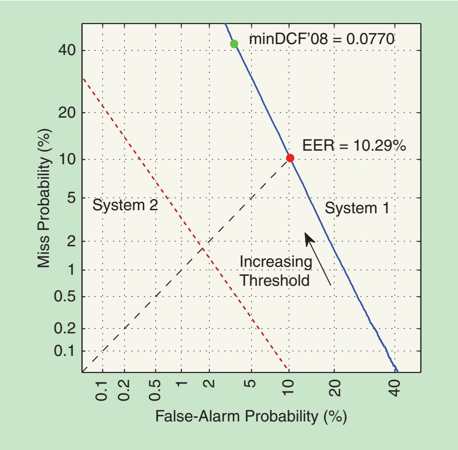
- Control the levels of these errors by setting decision threshold
- Equal error rate – FAR and FRR values when they are equal
- DET (detection error tradeoff) curve – plots FRR (miss probability) against FAR (false alarm probability)

Speaker verification decision threshold



Source: Hansen and Hasan, 2015

DET curve



Source: Hansen and Hasan, 2015

Detection cost function

- Detection cost function takes into account
  - Cost of miss (C_{miss}) and false alarm (C_{FA}) errors
  - Prior probability of target speaker – P_{target})
  - Miss probability at threshold τ – $P_{\text{miss}}(\tau)$
  - FA probability at threshold τ – $P_{\text{FA}}(\tau)$

$$DCF(\tau) = C_{\text{miss}} P_{\text{miss}}(\tau) P_{\text{target}} + C_{\text{FA}} P_{\text{FA}}(\tau)(1 - P_{\text{target}})$$

- Set $C_{\text{miss}} > C_{\text{FA}}$ if it is better to have false alarms than it is to miss the target speaker (e.g. law enforcement applicationa)

Features for speaker verification

- Frame-level – typically use MFCCs or other features used in ASR
- Utterance/speaker-level – since we require to make decisions at the utterance level often aim to learn utterance level representations or embeddings
  - GMM supervectors
  - i-vectors
  - DNN embeddings
    - d-vectors
    - x-vectors

GMM-based speaker verification

- UBM (Universal Background Model) – train a GMM with many Gaussians (eg 2048) on the speech of the general population
    - NB: no sequence modelling (no HMM) - just a distribution over MFCCs
- Then adapt the UBM to each target speaker using MAP adaptation
- Directly use these GMMs to verify a target speaker using the log likelihood ratio (LLR), where X is the observed test utterance, θ_s is the target speaker model, and θ_0 is the UBM. :

$$LLR(X, s) = \log \frac{p(X|\theta_s)}{p(X|\theta_0)} = \log p(X|\theta_s) - \log p(X|\theta_0)$$

For a threshold τ

- If $LLR(X, s) \geq \tau$ then accept
- If $LLR(X, s) < \tau$ then reject

Recap: MAP adaptation

- Basic idea MAP adaptation balances the parameters estimated on the universal data with estimates from the target speaker
- Consider the mean of the mth Gaussian, $\boldsymbol{\mu}_m$
  - ML estimate of SI model:

$$\boldsymbol{\mu}_m = \frac{\sum_n \gamma_m(n)\mathbf{x}_n}{\sum_n \gamma_m(n)}$$

    where $\gamma_m(n)$ is the component occupation probability

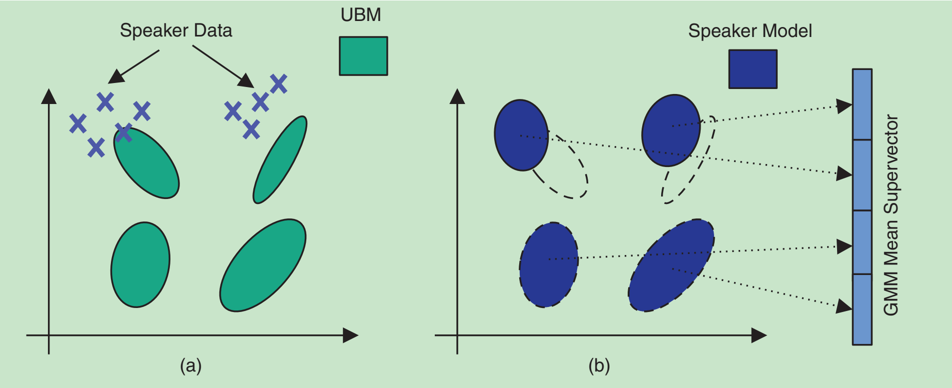
Recap: MAP adaptation

- Basic idea MAP adaptation balances the parameters estimated on the universal data with estimates from the target speaker
- Consider the mean of the mth Gaussian, $\boldsymbol{\mu}_m$
  - ML estimate of SI model:
  $$\boldsymbol{\mu}_m = \frac{\sum_n \gamma_m(n)\mathbf{x}_n}{\sum_n \gamma_m(n)}$$
  where $\gamma_m(n)$ is the component occupation probability
  - **MAP estimate** for the adapted model:
  $$\hat{\boldsymbol{\mu}} = \frac{\alpha\boldsymbol{\mu}_0 + \sum_n \gamma(n)\mathbf{x}_n}{\alpha + \sum_n \gamma(n)}$$
    - α controls balances the SI estimate and the adaptation data (typically $0 \leq \alpha \leq 20$)
    - $\mathbf{x}_n$ is the adaptation vector at time n
    - $\gamma(n)$ the probability of this Gaussian at this time

Recap: MAP adaptation

- Basic idea MAP adaptation balances the parameters estimated on the universal data with estimates from the target speaker
- Consider the mean of the mth Gaussian, $\boldsymbol{\mu}_m$
  - ML estimate of SI model:
    $$\boldsymbol{\mu}_m = \frac{\sum_n \gamma_m(n)\mathbf{x}_n}{\sum_n \gamma_m(n)}$$
    where $\gamma_m(n)$ is the component occupation probability
  - **MAP estimate** for the adapted model:
    $$\hat{\boldsymbol{\mu}} = \frac{\alpha\boldsymbol{\mu}_0 + \sum_n \gamma(n)\mathbf{x}_n}{\alpha + \sum_n \gamma(n)}$$
    - α controls balances the SI estimate and the adaptation data (typically $0 \leq \alpha \leq 20$)
    - $\mathbf{x}_n$ is the adaptation vector at time n
    - $\gamma(n)$ the probability of this Gaussian at this time
  - As the amount of training data increases, MAP estimate converges to ML estimate

GMM UBM system



Source: Hansen and Hasan, 2015

i-Vectors

- Represent a speaker using the GMM (mean) parameters – concatenate the target speaker mean parameters to form a **GMM supervector** $\boldsymbol{m}_s$. Typical dimension of a UBM GMM is 2048, so with 39-dimension parameters, this can be a very high dimension vector ($\sim 80,000$ components)
- Represent the supervector for an utterance $\boldsymbol{X}_u$ as the combination of the UBM supervector and the utterance **i-vector** (Dehak et al, 2011):

$$\boldsymbol{m}_u = \boldsymbol{m}_0 + \boldsymbol{T}\boldsymbol{w}_u$$

  - $\boldsymbol{m}_u$ and $\boldsymbol{m}_0$ are D-dimension supervectors for the utterance u and the UBM
  - $\boldsymbol{w}_u$ is the **i-vector** ("identity vector") – a reduced dimension (d) representation for utterance u ($d \sim 400$)
  - $\boldsymbol{T}$ is a $D \times d$ matrix (sometimes called the "total variability matrix") which projects the supervector down to the i-vector representation
  - Estimate $\boldsymbol{T}$ for the development corpus using an EM algorithm, estimate the i-vector $\boldsymbol{w}_u$ for an utterance as the mean of the (Gaussian) posterior distribution of $\boldsymbol{w}_u$ given $\boldsymbol{X}_u$ and $\boldsymbol{T}$.

Speaker verification scoring using i-vectors

- Speaker verification involves computing a score $f(\boldsymbol{w}_{\text{target}}, \boldsymbol{w}_{\text{test}})$ between the target and test i-vectors
- Cosine score

$$f_{\cos}(\boldsymbol{w}_{\text{target}}, \boldsymbol{w}_{\text{test}}) = \frac{\boldsymbol{w}_{\text{target}} \cdot \boldsymbol{w}_{\text{test}}}{||\boldsymbol{w}_{\text{target}}|| \, ||\boldsymbol{w}_{\text{test}}||}$$

- Probabilistic linear discriminant analysis (PLDA) – probabilistic model that accounts for speaker variability and channel variability. Can be used to compute the log likelihood ratio, so

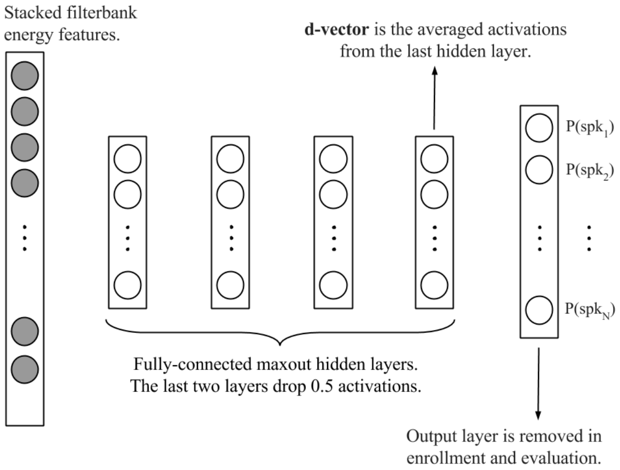
$$f_{\text{plda}}(\boldsymbol{w}_{\text{target}}, \boldsymbol{w}_{\text{test}}) = \log p(\boldsymbol{w}_{\text{target}}, \boldsymbol{w}_{\text{test}} | H_1) - \log \left[p(\boldsymbol{w}_{\text{target}} | H_0) p(\boldsymbol{w}_{\text{test}} | H_0) \right]$$

  where H_1 is the hypothesis that the test and target speakers are the same, H_0 is the hypothesis they are different
- PLDA is current-state of the art for scoring i-vectors

Neural network approaches

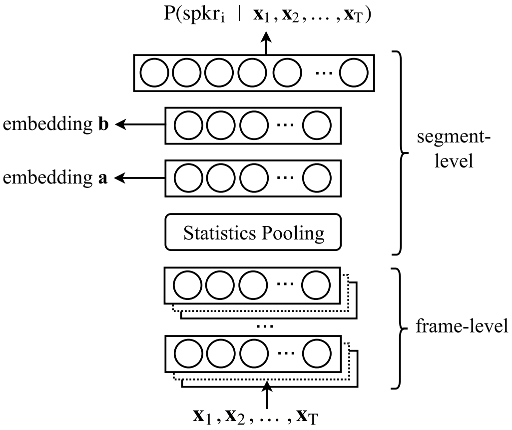
- Current state-of-the-art neural network approaches use NNs to extract embeddings, which are then scored by PLDA
- d-vectors (Variani et al, 2014)
  - Development – train a DNN to recognise speakers
  - Enrolment – extract speaker-specific features from last hidden layer
  - d-vector – average speaker-specific features across frames of an utterance (pooling)
- x-vectors (Snyder et al, 2018)
  - Similarly to d-vectors extract an utterance level feature as an embedding
  - Train TDNN with frame-level input and utterance-level output
  - Architecture includes a "stats pooling" layer which computes mean and sd across the utterance of the highest frame-level hidden layer

d-vector extraction



Source: Variani et al, 2014

x-vector extraction



Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

Source: Snyder et al, 2018

Summary

- i-vectors are the state-of-the-art speaker representation, used in
  - speaker recognition
  - speaker diarization
  - speaker adaptation in ASR
- NN speaker representations such as d-vectors and x-vectors are now competitive with i-vectors
- PLDA is the state-of-the-art scoring approach
- Current challenges include development of end-to-end NN approaches

Reading

- JHL Hansen and T Hasan (2015), “Speaker Recognition by Machines and Humans: A tutorial review”, *IEEE Signal Processing Magazine*, 32(6):74–99, https://ieeexplore.ieee.org/document/7298570
- MW Mak and JT Chien (2016), “Tutorial on Machine Learning for Speaker Recognition”, Interspeech, http://www.eie.polyu.edu.hk/~mwmak/papers/IS2016-tutorial.pdf
- N Dehak et al (2011), “Front-End Factor Analysis for Speaker Verification”, *IEEE Trans Audio, Speech, and Language Processing*, 19(4):788–798, https://ieeexplore.ieee.org/document/5545402
- E Variani et al (2014), “Deep neural networks for small footprint text-dependent speaker verification”, ICASSP, https://ieeexplore.ieee.org/document/6854363
- D Snyder et al (2018), “X-Vectors: Robust DNN Embeddings for Speaker Recognition”, ICASSP, https://ieeexplore.ieee.org/document/8461375