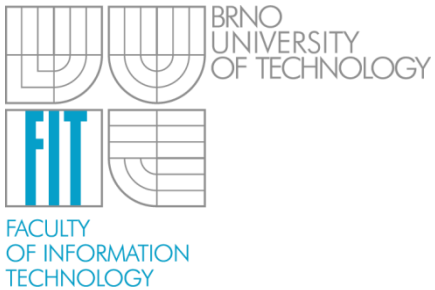
Speaker Recognition

Oldřich Plchot
Speech@FIT, Brno University of Technology, Czech Republic
iplchot@fit.vutbr.cz

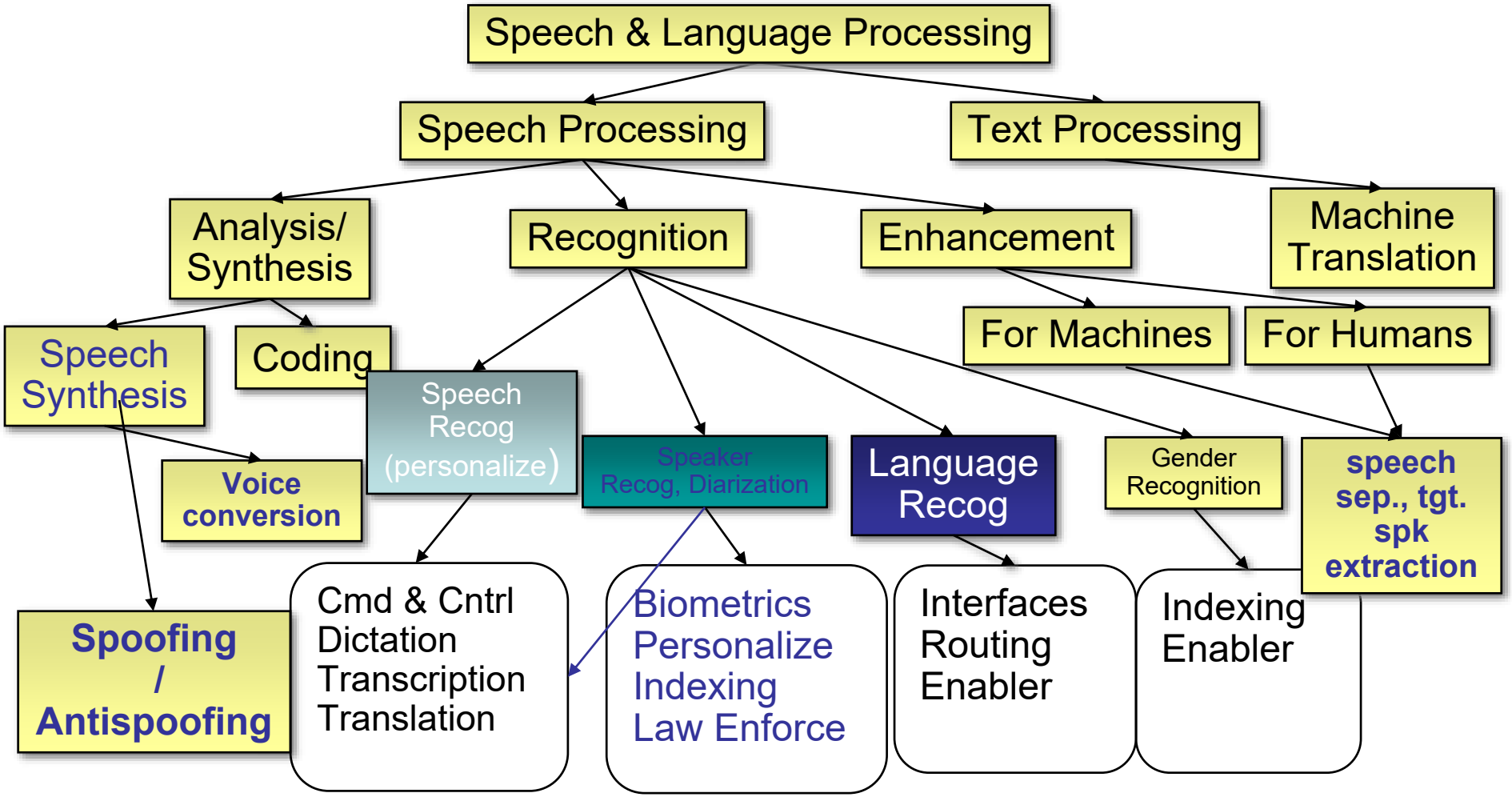


ZRE
Brno
5th May, 2025

Agenda

- Speaker recognition applications
  - Why we need speaker recognition
  - Different application domains
- Background and theory
  - Feature extraction
  - Gaussian Mixture Modelling, i-vectors
  - Modelling of embeddings, PLDA
  - Current trends in speaker verification
- Evaluation metrics
- Score calibration and normalization
- Conclusions

Speech processing, use of speaker representations



Speaker Recognition Areas of Use

Security and defense
Forensic
Looking for suspect in quantity of audio
Waiting online for suspect

Access Control
Physical facilities
Computer networks & websites

Transaction Authentication
Telephone banking
Remote purchases

Speech Data Management
Voice mail browsing
Search in audio archives

Personalization
Voice-web/device customization
Intelligent answering machine

Biometrics

- **Biometric**: a human generated signal or attribute for authenticating a person's identity
- Voice biometric can be combined with other forms to reach highest security
  - Voice +
  - Finger print
  - Face, Iris

- **Voice is popular biometric**
  - Natural signal to produce
  - Does not require a specialized input device
  - Can be obtained remotely, without speaker assistance

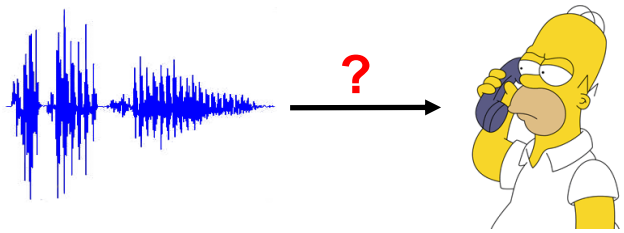
Speaker Recognition Tasks

Identification

Whose voice is this?

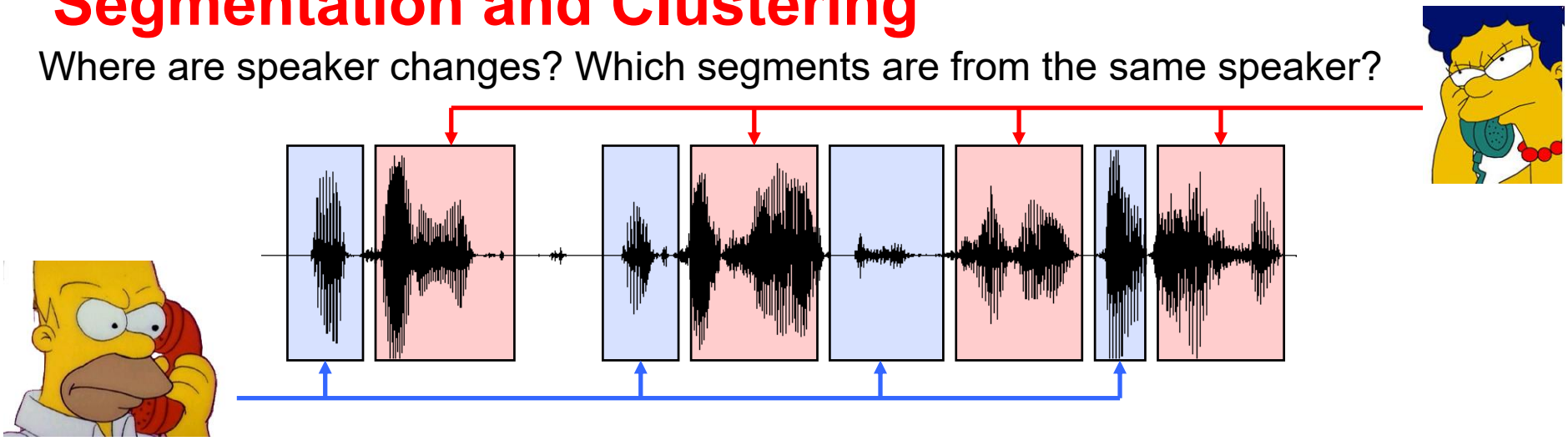
Verification

Is this Homer's voice?



Segmentation and Clustering

Where are speaker changes? Which segments are from the same speaker?



Pure biometrics vs combination with "password"

Different applications are suited for different approaches

Text-dependent

- Recognition system knows text spoken by person
- Example: fixed or prompted phrases
- Used for applications with strong control over user
- Knowledge of spoken text can improve system performance

Text-independent

- Recognition system does not know text spoken by person
- Example: User selected phrases, conversational speech
- Used for applications with less or no control over user
- More flexible system but more difficult problem

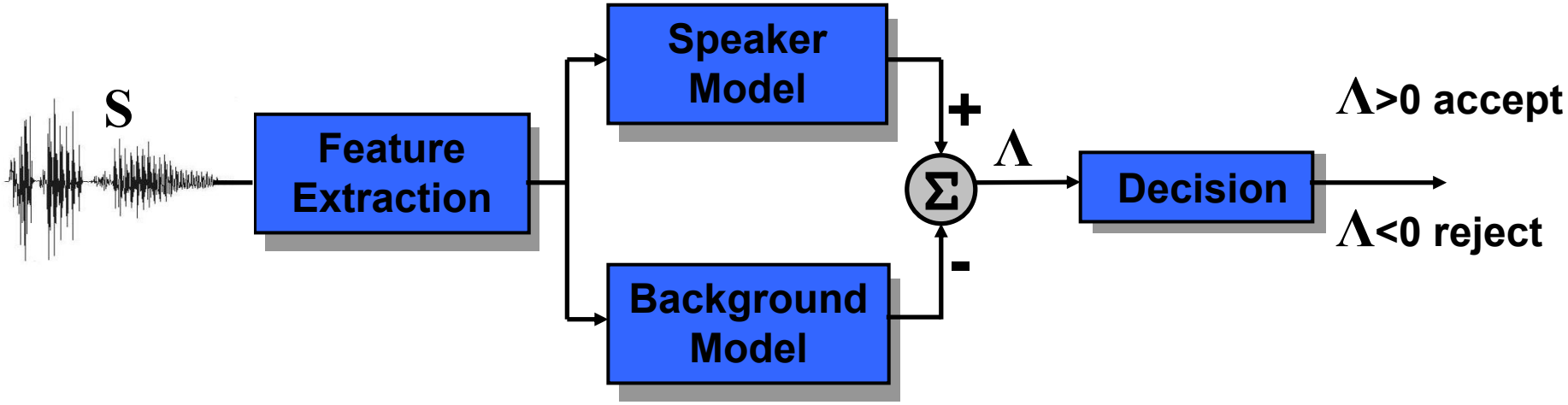
Agenda

- Speaker recognition applications
  - Why we need speaker recognition
  - Different application domains
- Background and theory
  - Feature extraction principles
  - Gaussian Mixture Modelling, i-vectors
  - Modelling of embeddings, PLDA
- Evaluation metrics
- Score calibration and normalization
- Conclusions

Speaker detection decision approaches have roots in signal detection theory

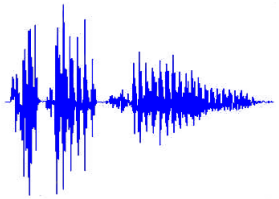
- **2 class Hypothesis test**
  - H0: the speaker is **not** the target speaker
  - H1: the speaker is the target speaker
- **Statistic computed on test utterance S as likelihood ratio**

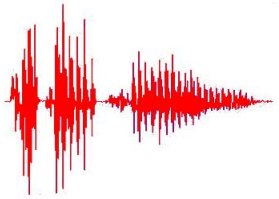
$$\Lambda = \log \frac{\text{Likelihood } \mathbf{S} \text{ came from speaker model}}{\text{Likelihood } \mathbf{S} \text{ did } \underline{\mathbf{not}} \text{ come from speaker model}}$$



Two distinct phases to any speaker detection system

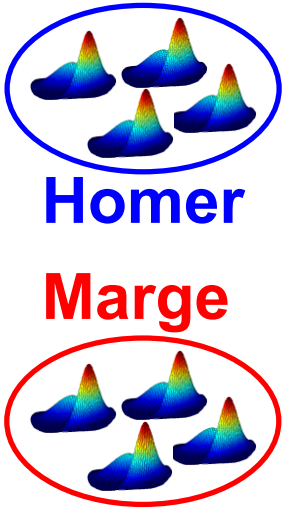
Training phase



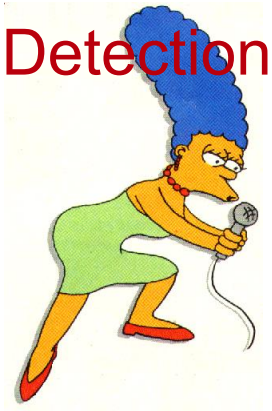


Feature Extraction → Model Training → Homer

Marge



Detection phase



Feature Extraction → Detection Decision → Detected

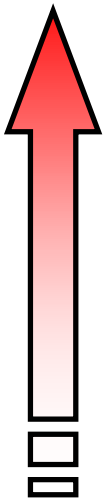
Hypothesized identity - **Marge**

Features for Speaker Recognition

- Humans use several levels of perceptual cues for speaker recognition

Hierarchy of Perceptual Cues

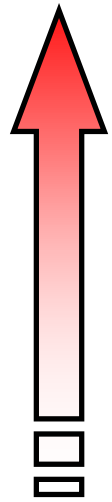
High-level cues (learned traits)



Low-level cues (physical traits)

Semantics, idiolect, pronunciations, idiosyncrasies	Socio-economic status, education, place of birth
Prosodics, rhythm, speed intonation, volume modulation	Personality type, parental influence
Acoustic aspects of speech, nasal, deep, breathy, rough	Anatomical structure of vocal apparatus

Difficult to automatically extract



Easy to automatically extract

- There are no exclusive speaker identity clues

Features for Speaker Recognition

- Desirable attributes of features for an automatic system (Wolf 72)

Practical

- **Occur naturally and frequently in speech**
- **Easily measurable**

Robust

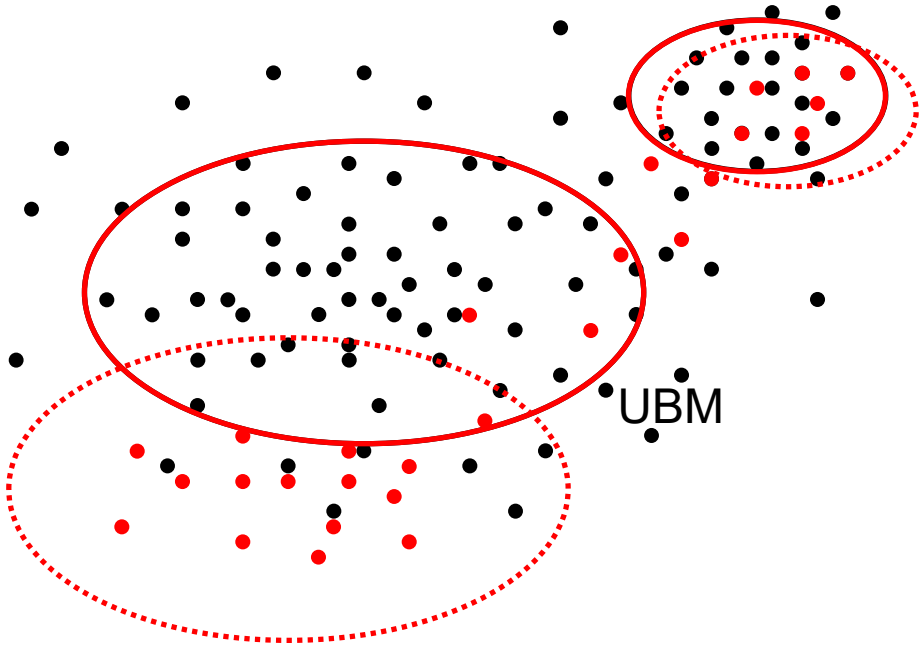
- **Not change over time or be affected by speaker health**
- **Not be affected by reasonable background noise nor depend on specific transmission characteristics**

Secure

- **Not be subject to mimicry**

- No features have all these attributes
- Features derived from spectrum of speech have proven to be the most effective in automatic systems
  - Mel-filter bank outputs, MFCCs and other variants, raw signal processed by trained CNN (transformers)

MAP adaptation – How to create speaker model



- Target speaker data

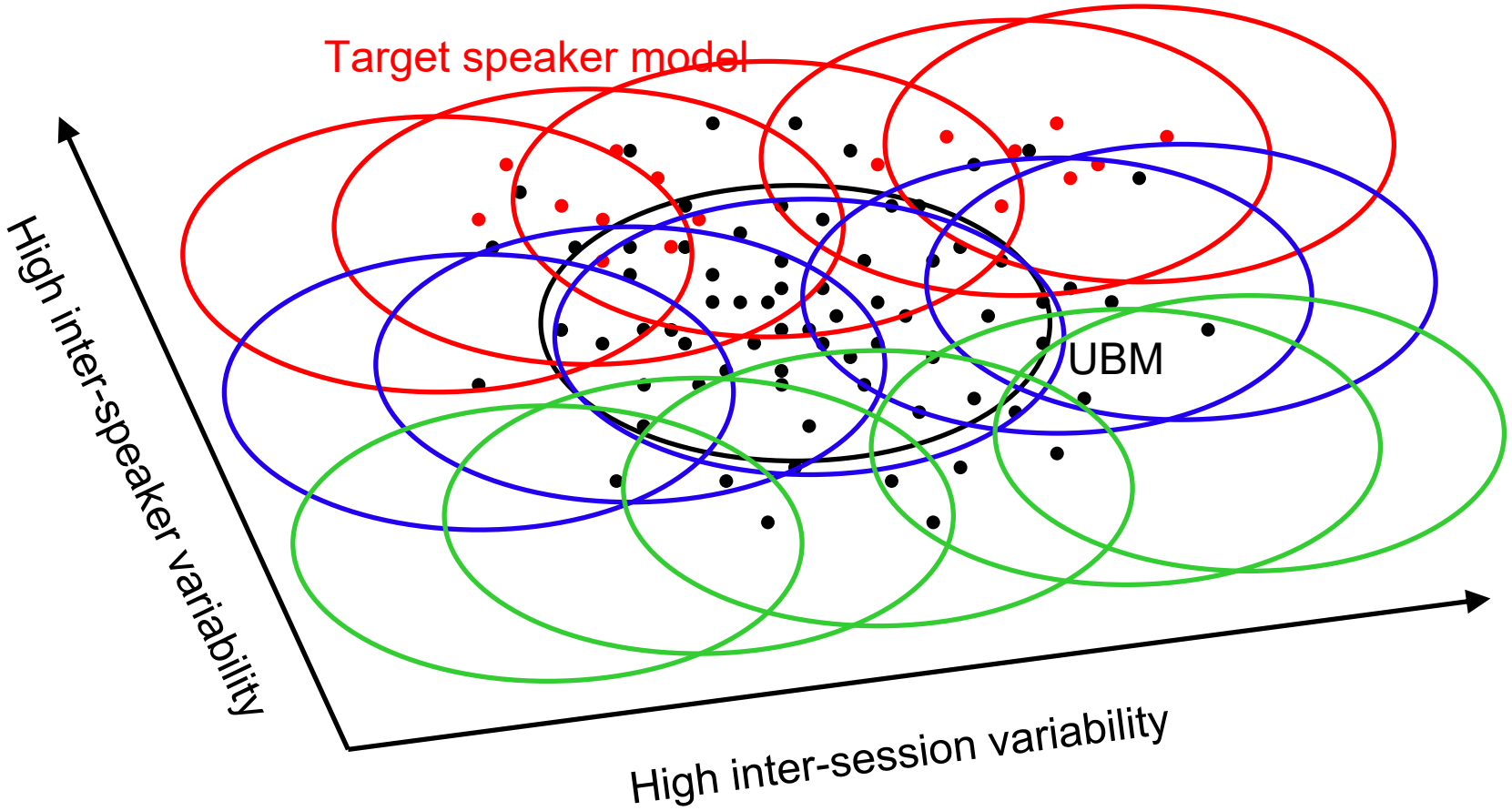
- UBM model - 2 Gaussians
- Speaker model adapted from UBM
- Adapted only parameters seen in target training data
- Only means adapted
- $\mu_{tgt} = \gamma\ \mu_{trn} + (1-\gamma)\ \mu_{ubm}\ ;\ \gamma = n/(n+r)$
- n … number of target training data
- r … relevance factor – for speaker id. usually 10-19

Channel/session effects

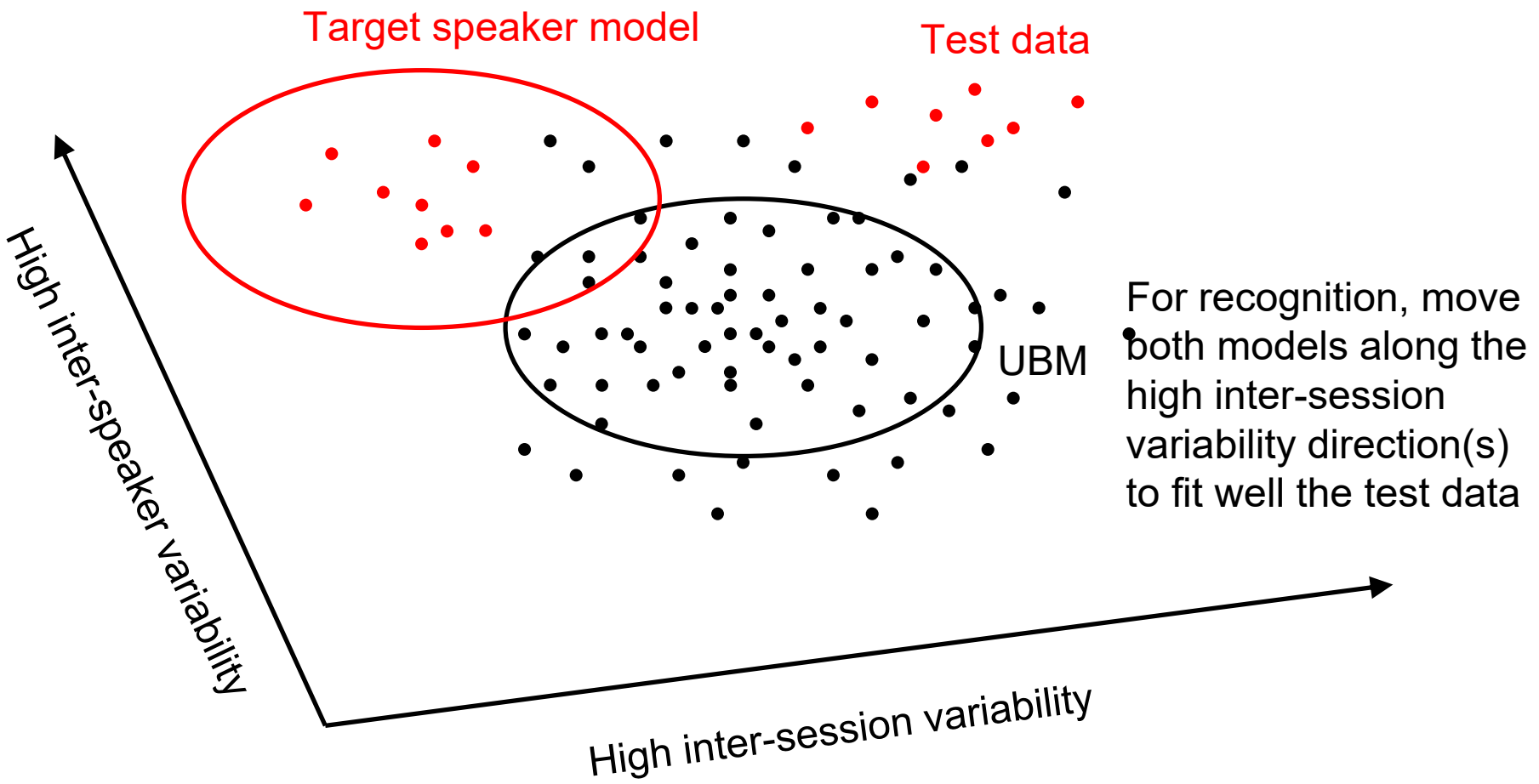
The largest challenge to practical use of speaker detection systems is channel/session variability

- **Variability** – refers to changes in channel between training and successive detection attempts
- **Channel/session** effects encompasses several factors
  - The microphones
    Carbon-button, electret, hands-free, array, …
  - The acoustic environment
    Office, car, airport, street, restaurant, …
  - The transmission channel
    Landline, cellular, VoIP,…
  - Speaker emotion state
    Calm, nervous, stress, drunk, ill, …

Inter-session variability



Inter-session variability compensation



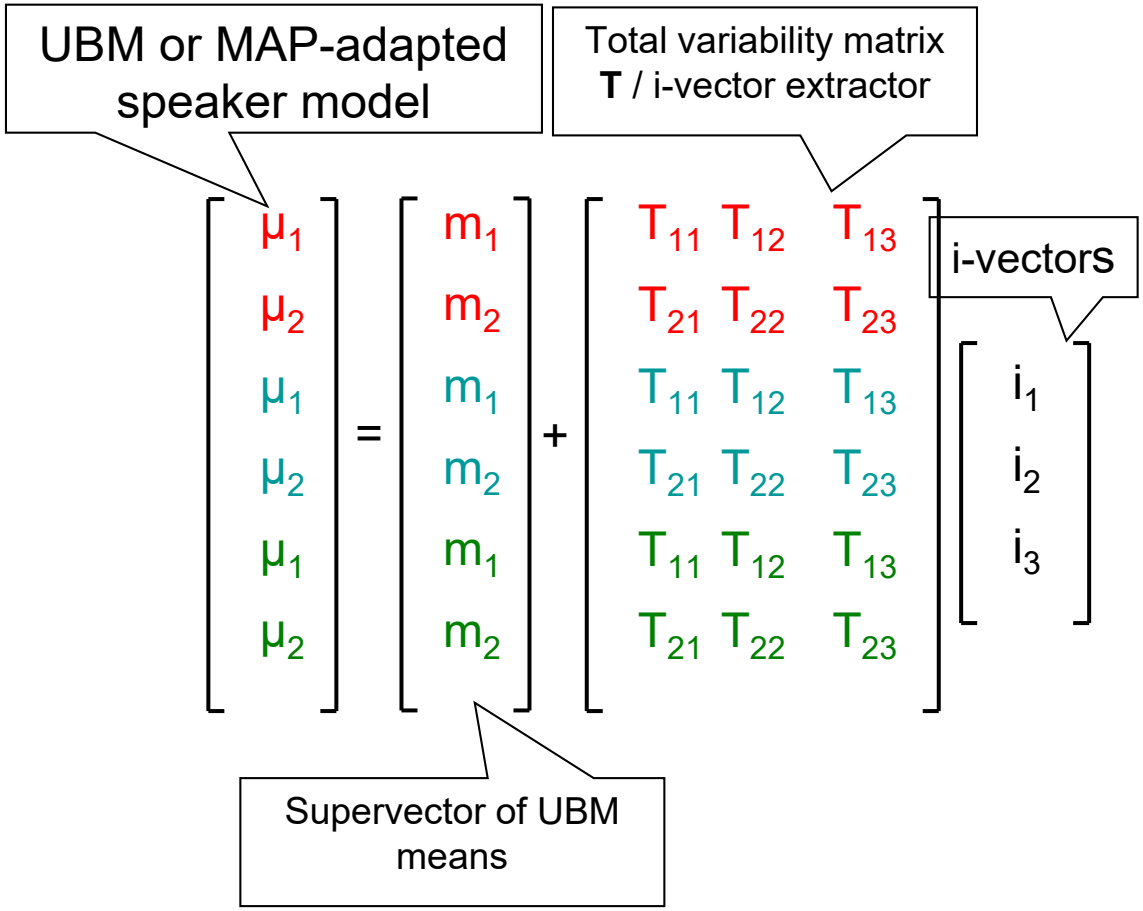
Transition to i-vectors

- **Eigen-{voice,channel} adaptation – compensating for speaker and channel variability, moving the model parameters in the subspace of given variability**
- **Everything happens in high-dimensional space of GMM mean supervectors…**
- **Both techniques identify the low-dimensional subspace(s) which are important for modeling**

Use only the notion of representing the recording by fixed-length low-dimensional vector => i-vectors

- **Channel compensation still to be done in the i-vector space.**

- We maximize the overall likelihood of the data given the subspace model via EM algorithm and find the i-vector extractor matrix
- No labels are needed during training
- During test, we compute the MAP point estimate of **x** given the fixed extractor



iVectors – low-dimensional recording representation

$$\tilde{\mu} = \mathbf{m} + \mathbf{T}\mathbf{i}$$

- The main idea is to transform large utterance-specific GMM supervector into a small subspace while retaining most of the variability
- iVector extractor is model similar to JFA
  - with single subspace $\mathbf{T}$ -> easier to train
  - no need for speaker labels -> the subspace can be trained on large amount of unlabeled recordings
- low-dimensional (typically 200-600 dimensions)
- i-vector is information rich
  - contains information about speaker, language, emotion, … and also channel -> **needs to be considered by the following classifier**.
- We assume standard normal prior for factors $\mathbf{i}$.
- **Estimate the i-vector -- MAP point estimate of i, for every recording**

Dehak, N., et al., Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification, In Proc Interspeech 2009, Brighton, UK, September 2009

Probabilistic Linear Discriminant Analysis (PLDA)

- What would be now the appropriate probabilistic model for verification?
  - i-vector still contains channel information -> our model should consider both speaker and channel variability.
- Such model is known as PLDA and is described by familiar equation:

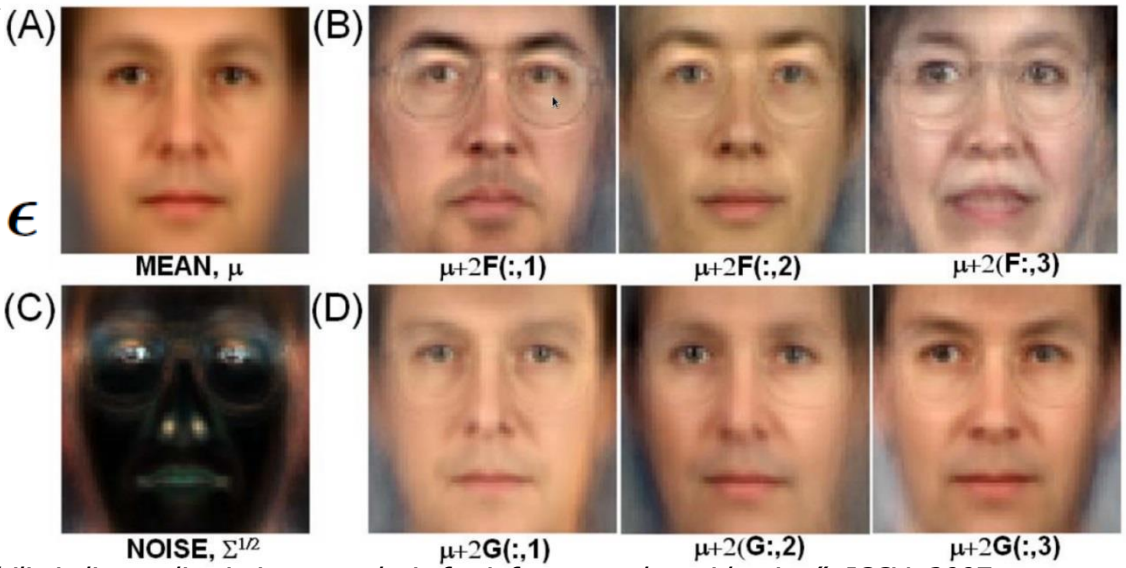
$$\mathbf{i} = \boldsymbol{\mu} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x} + \boldsymbol{\epsilon}$$

- This equation now describes directly the variability of observed data – i-vectors (generative embeddings)
- Parameters $\boldsymbol{\mu}$, $\mathbf{V}$, $\mathbf{U}$ and diagonal covariance matrix of residual noise vector $\boldsymbol{\epsilon}$ can be estimated using EM algorithm.

PLDA continued

- PLDA has nice interpretation in face verification where it was introduced by Simon J.D. Prince
- Each face image $\mathbf{i}$ can be constructed by adding
  - (A) mean face $\boldsymbol{\mu}$
  - (B) linear combination of basis $\mathbf{V}$ corresponding to between-individual variability (moving from $\boldsymbol{\mu}$ in these directions gives us images that look like different people)
  - (D) linear combination of basis $\mathbf{U}$ corresponding to within-individual variability (moving from $\boldsymbol{\mu}$ in these directions gives us images that look like the same person under different condition – e.g. lighting)
  - (C) residual noise vector $\boldsymbol{\epsilon}$

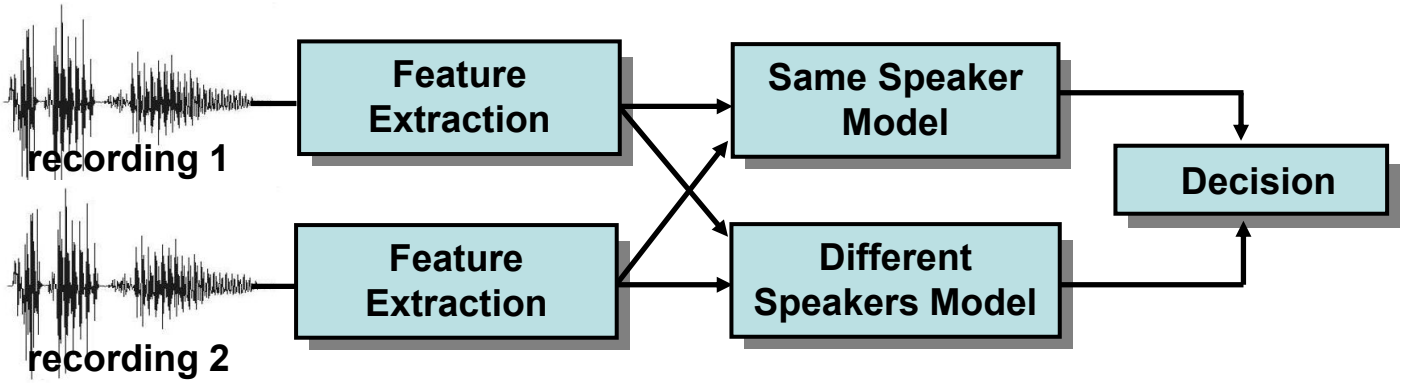
$$\mathbf{i} = \boldsymbol{\mu} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x} + \boldsymbol{\epsilon}$$



Picture taken from: S.J.D. Prince and J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," ICCV, 2007

PLDA based verification

- Verification based on Bayesian model comparison
- Compare likelihoods for two hypothesis:
  - H_s – both recordings come from the same speaker
  - H_d – recordings come from different speakers



- It is symmetrical problem
- We do not need to train explicitly the speaker model and compare to each test recording which was done before
- Scoring with PLDA is FAST!!

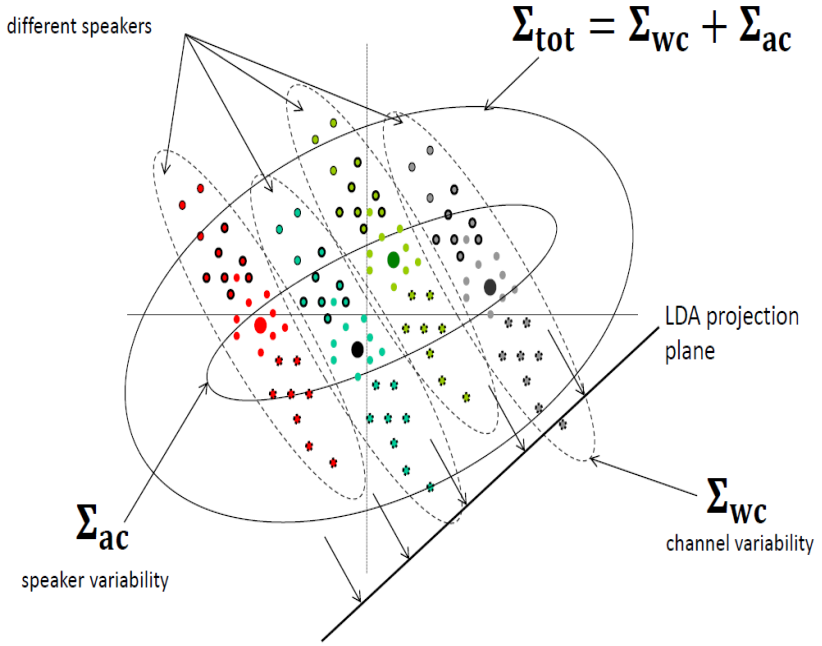
Simplified PLDA model

Labelled training data were required only for the probabilistic "backend" (typically PLDA).

This was one of a big advantages of i-vectors over its predecessor (Joint Factor Analysis).

The verification score is a log likelihood ratio of the utterances being generated jointly from the same speaker or independently from different speakers

$$s = \log \frac{l\left(\mathcal{X}_{e_1} \ldots \mathcal{X}_{e_n}, \mathcal{X}_{t_1} \ldots \mathcal{X}_{t_m} | H_s\right)}{l\left(\mathcal{X}_{e_1} \ldots \mathcal{X}_{e_n} | H_s\right) l\left(\mathcal{X}_{t_1} \ldots \mathcal{X}_{t_m} | H_s\right)}$$



Embeddings for Speaker Recognition

- David Snyder et al., INTERSPEECH 2017

Deep Neural Network Embeddings for Text-Independent Speaker Verification

David Snyder, Daniel Garcia-Romero, Daniel Povey, Sanjeev Khudanpur

Center for Language and Speech Processing & Human Language Technology Center of Excellence,
The Johns Hopkins University, USA

{david.ryan.snyder, dpovey}@gmail.com, {dgromero, khudanpur}@jhu.edu

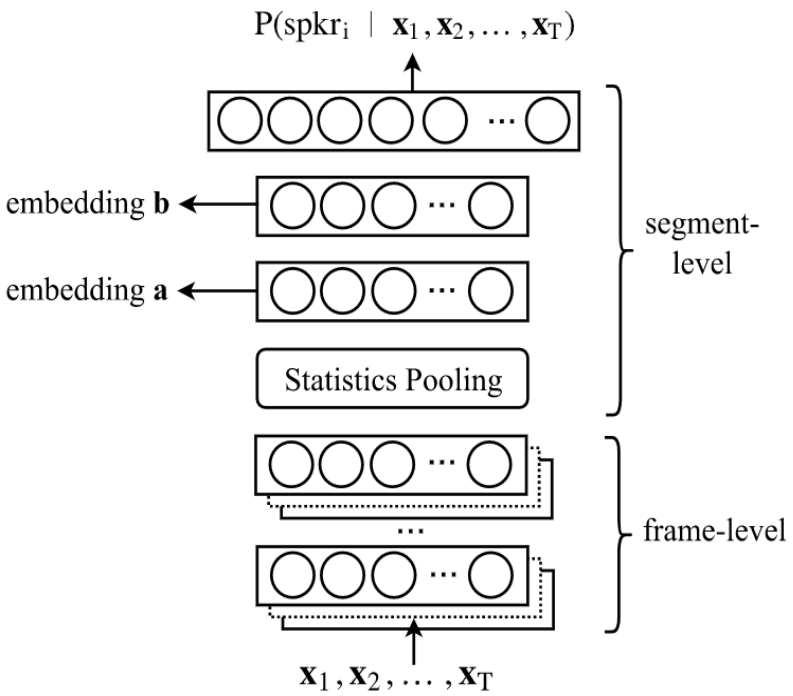
Abstract

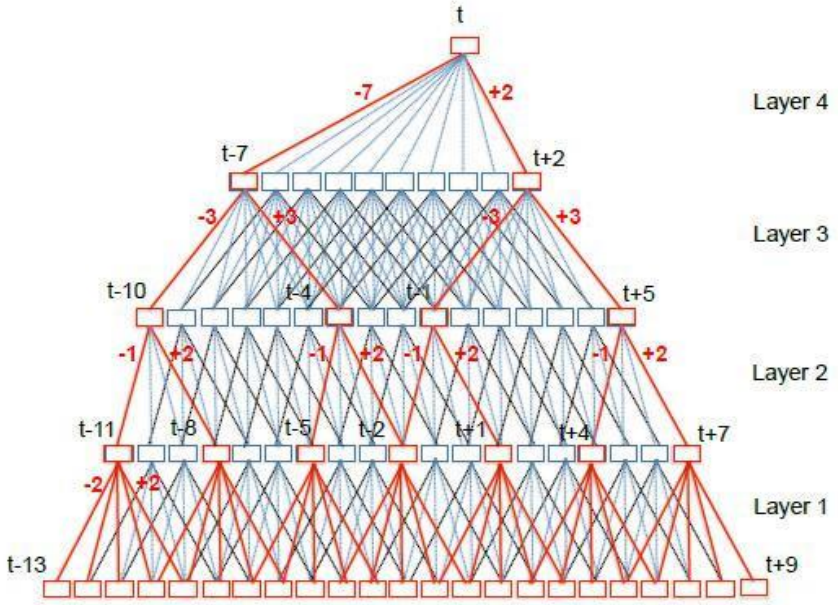
This paper investigates replacing i-vectors for text-independent speaker verification with embeddings extracted from a feed-forward deep neural network. Long-term speaker characteristics are captured in the network by a temporal pooling layer that aggregates over the input speech. This enables the network

model (UBM) that is used to collect sufficient statistics, a large projection matrix to extract i-vectors, and a probabilistic linear discriminant analysis (PLDA) backend to compute a similarity score between i-vectors [2, 3, 4, 5, 6, 7].

Traditionally, the UBM is a Gaussian mixture model (GMM) trained on acoustic features. Recent work has shown

Speaker-discriminative DNNs, x-vectors





gure 1: Computation in TDNN with sub-sampling (red) and ithout sub-sampling (blue+red)

https://blog.csdn.net/qq_14962179

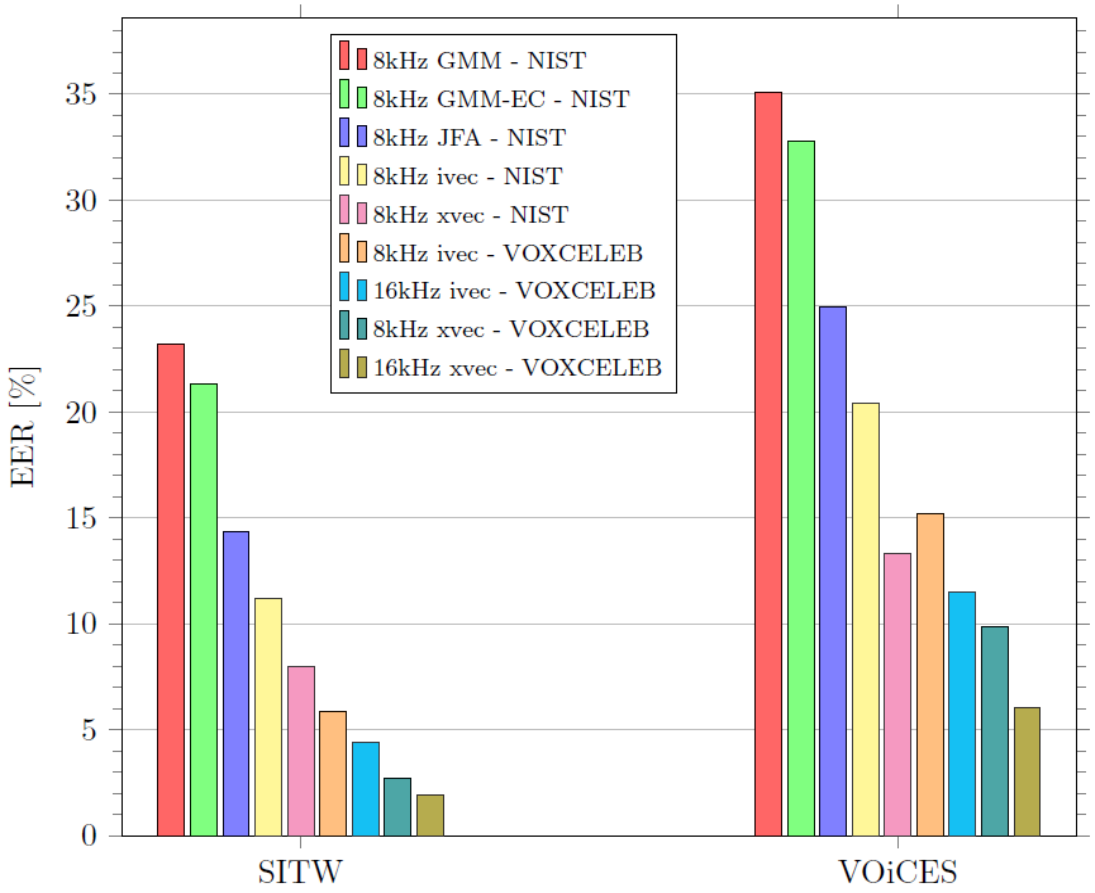
7 layers of Time-delay architecture (TDNN), Stats: mean and std, Embeddings: 512 and 300, Softmax output over supervised set of training speakers (few thousands), Trained with samples between 2 and 10 seconds

Peddinti, V., Povey, D., Khudanpur, S., *"A time delay neural network architecture for efficient modeling of long temporal contexts"* Proc. Interspeech 2015

Snyder, D., Garcia-Romero, D., Povey D., Khudanpur S. *"Deep Neural Network Embeddings for Text-Independent Speaker Verification"* , Interspeech 2017

Longitudinal Analysis

- Both SITW and Voices are 16K
- 2000 GMM-UBM
- 2006 GMM-EC
- 2008 JFA
- 2010 iVectors
- 2017 x-vectors
- Effect of in- vs out-domain data
- 8K vs 16K

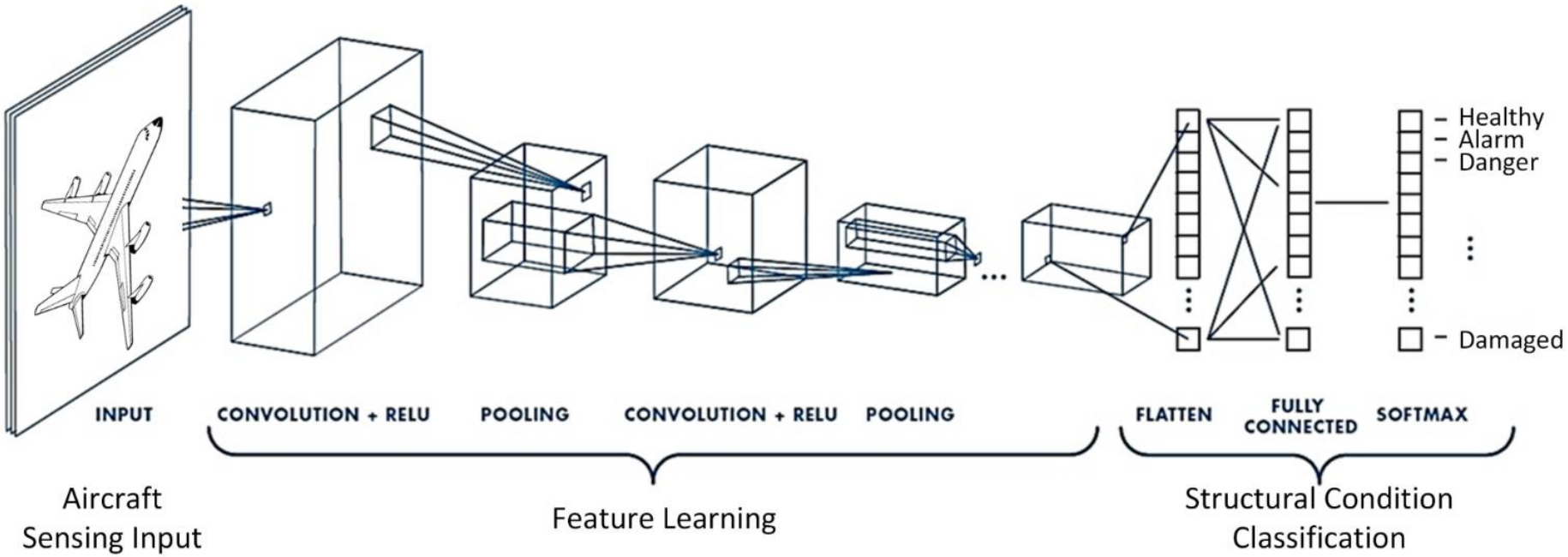


Pavel Matějka, et al., *"13 years of speaker recognition research at BUT, with longitudinal analysis of NIST SRE"*, Computer Speech & Language, vol. 63, 2020

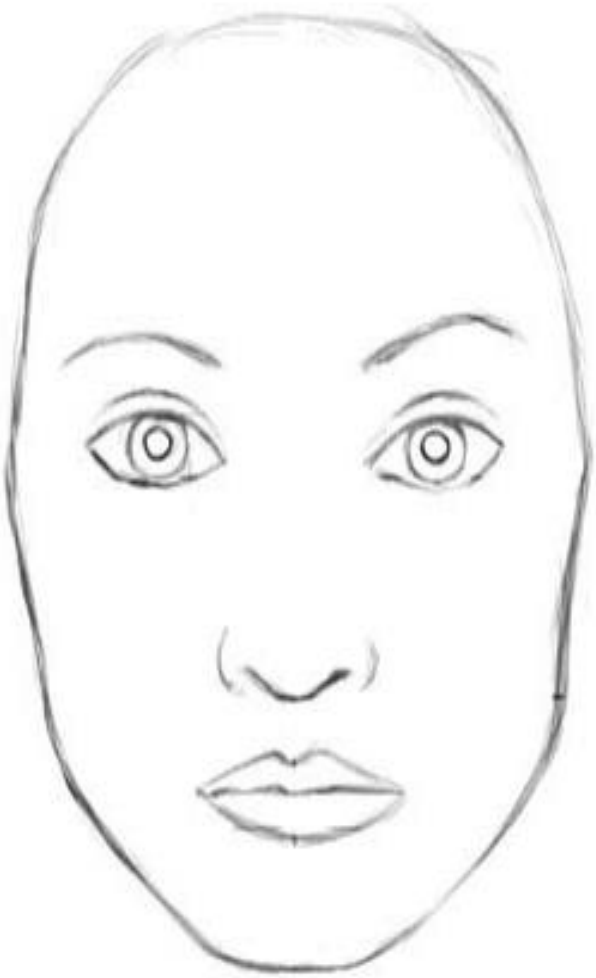
SOTA AND CURRENT RESEARCH DIRECTIONS

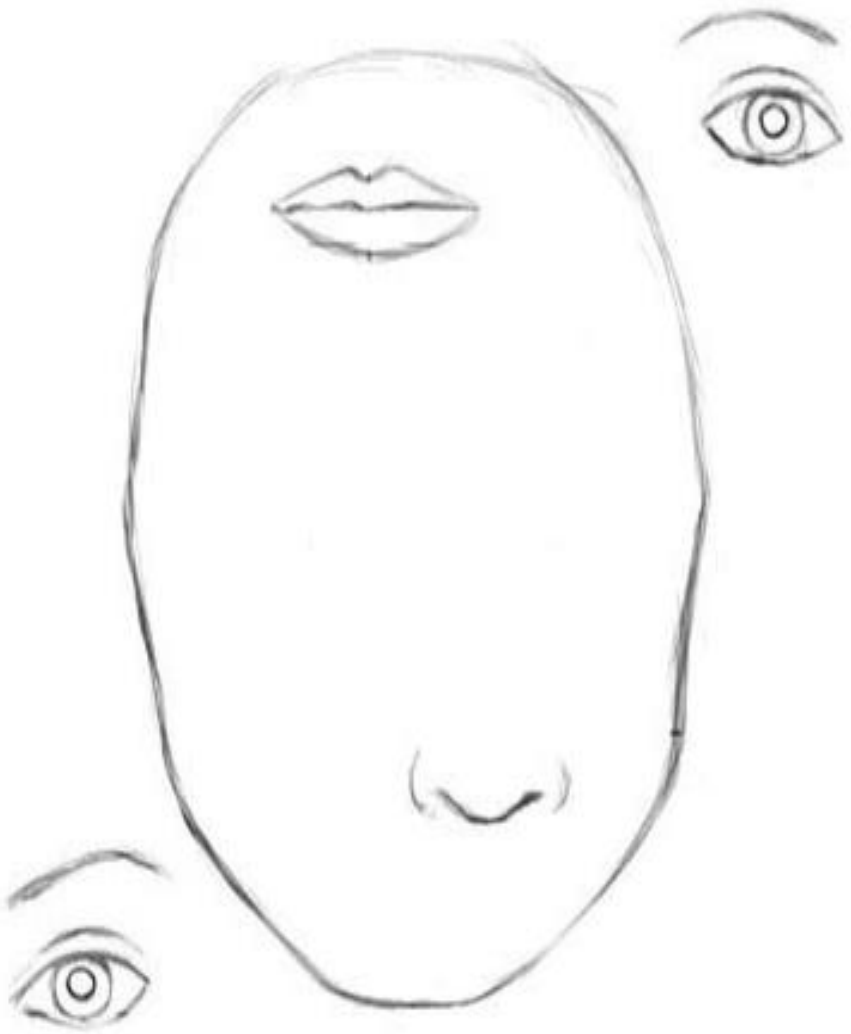
SOTA approach for feature learning - CNN

X-vectors with variants of large convolutional networks like ResNet, ECAPA-TDNN, RepVGG….
For coherent domains with large amounts of annotated data such as VoxCeleb
Voxceleb1-O is essentially solved, EERs << 1%

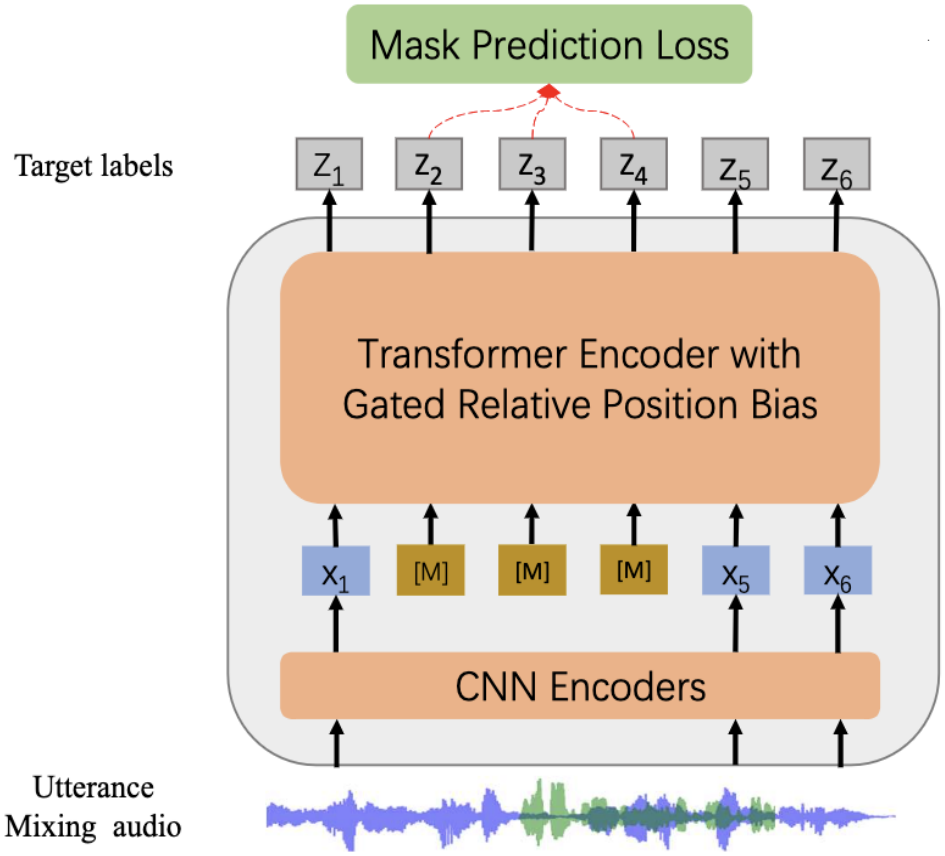


Attention vs CNN





WavLM

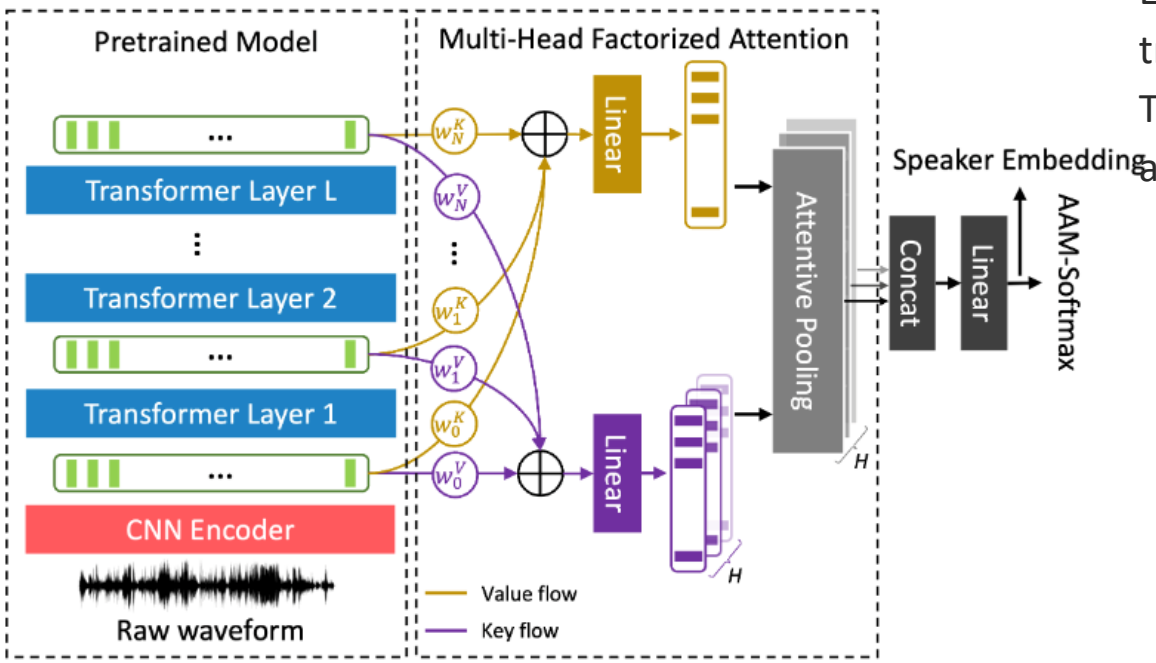


WavLM main characteristics

- Tailored to speaker recognition, diarization, speech separation, a.o.
- Jointly learns masked speech prediction and denoising.
- Simulate noisy/overlapped speech as inputs, and predict the pseudo-labels of original speech on the masked region.
- Replaces standard positional embeddings with gated relative position bias.

Chen, Sanyuan, et al. "WavLM: Large-scale self-supervised pre-training for full stack speech processing." (Microsoft) 2021

Attention-based SV backend on top of WavLM



Simple backend with multihead attention (64 heads).

Each head models an acoustic area via a trainable query vector.

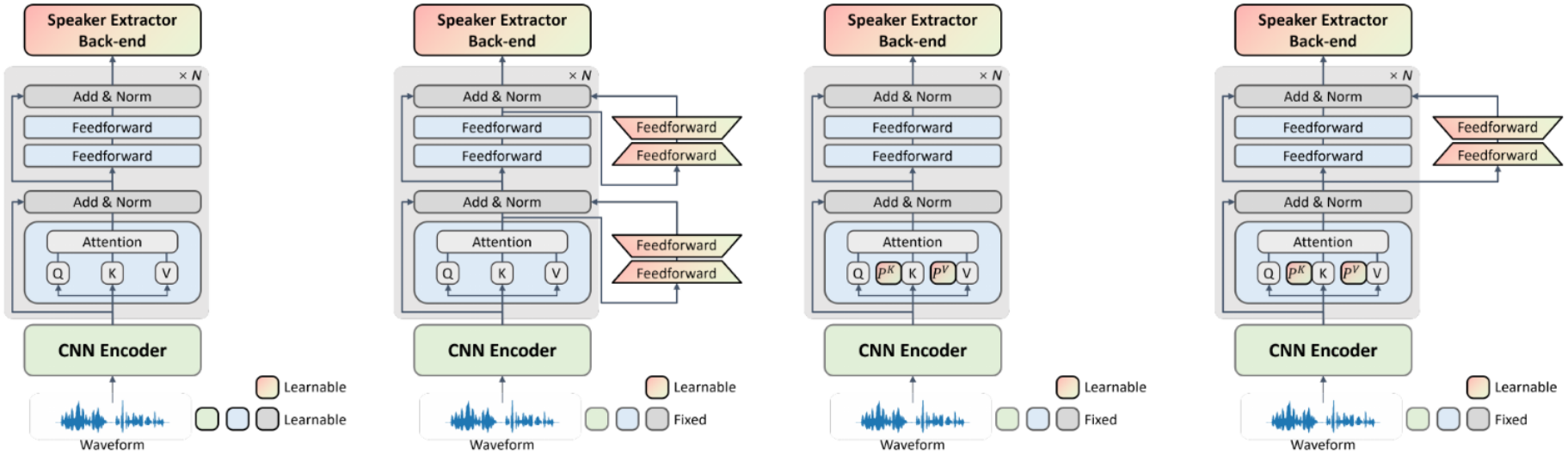
Two different layer-pooling weights (keys and values).

- Values should contain speaker information.
- Keys should not.

Peng, Junyi, et al. "An attention-based backend allowing efficient fine-tuning of transformer models for speaker verification." arXiv preprint arXiv:2210.01273 (2022)., SLT 2022

Transformer models and adapters for SV

- Do we need to finetune the whole SSL network?
- What if there is not enough in-domain data. (Experiments on VoxCeleb and CNCeleb)
- Interesting implications for concurrent deployment of many domain-specific models



(a) Full Fine-tuning (b) Bottleneck Adapter (c) Prefix Tuning (d) Mix-and-Match Adapter

Peng, Junyi, et al. ***"Parameter-efficient transfer learning of pre-trained Transformer models for speaker verification using adapters."*** *IEEE-ICASSP 2023*

- **Not enough labelled data**
  - Utilize pre-training paradigm that leverages vast amount of unlabeled data (or download the model)
  - Pre-trained model can be easily fine-tuned for target application (domain)
  - In the end, less labeled data are needed w.r.t. CNN or RNN-based models
- **Plenty of labelled data**
  - Train large CNN-based supervised embedding extractors
  - Obtain SOTA results, but perhaps lose some robustness

Agenda

- Speaker recognition applications
  - Why we need speaker recognition
  - Different application domains
- Background and theory
  - Feature extraction
  - Gaussian Mixture Modelling
  - Modelling of embeddings, PLDA
  - DNN embeddings

- Evaluation metrics
- Score calibration and normalization
- Conclusions

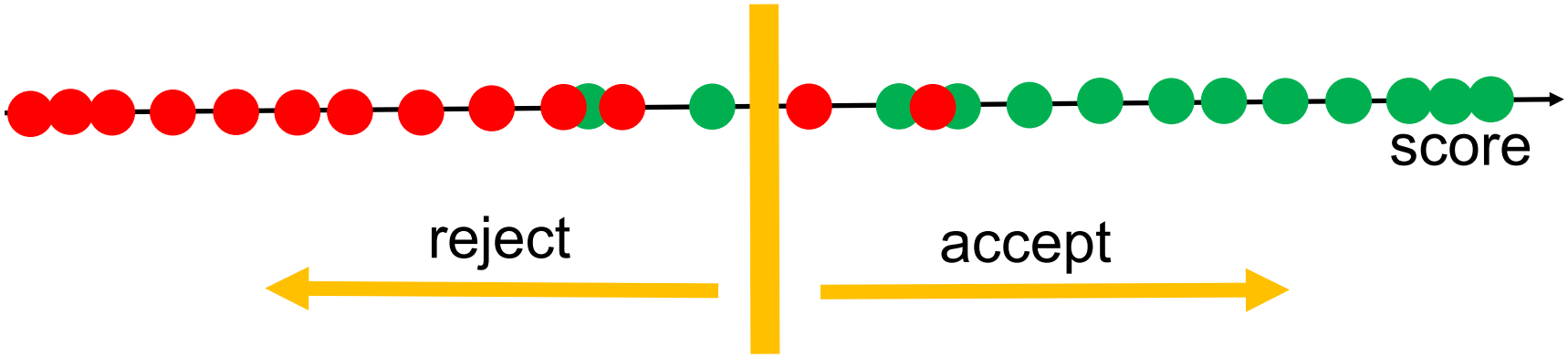
Evaluation Metric

- There are two types of errors
  - **MISS**: incorrectly rejected a target trial
    - It was target trial, but system said it is not
    - also known as a false reject
  - **FALSE ALARM**: incorrectly accept a non-target trial
    - It was not target trial, but system said it is
    - also known as a false accept
- The performance of a detection system is measure of the trade-off between these two errors – is controlled by adjustment of the decision threshold
- In an evaluation, N_{target} target-trials (test speaker = model speaker) and $N_{non\text{-}target}$ non-target trials (test speaker != model speaker) are conducted and error probabilities are estimated at given threshold

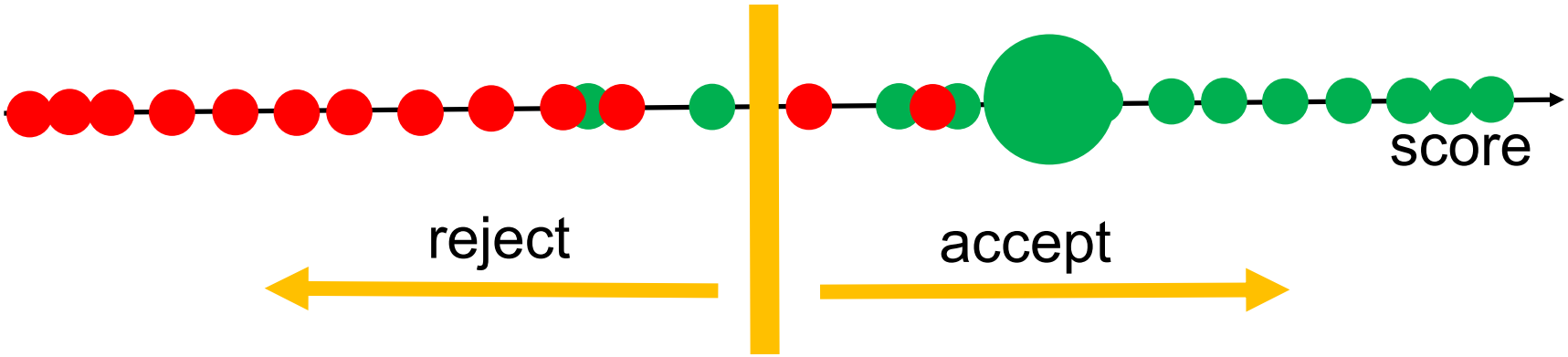
$$\Pr(\text{miss}|\text{thr}) = \frac{\#\text{ target trials scores} < \text{thr}}{N_{target}} \qquad \Pr(\text{false alarm}|\text{thr}) = \frac{\#\text{ non-target trials scores} > \text{thr}}{N_{non\text{-}target}}$$

Does the system work well ?

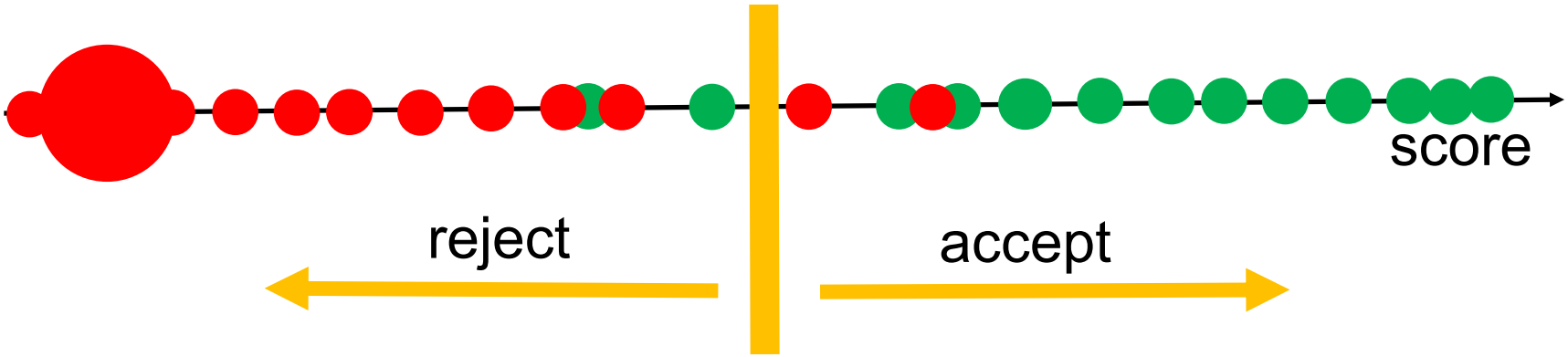
- We need some (lots of) data – pairs model speaker – test speaker (we call these pairs trials).
  - **Target-trials** (test speaker = model speaker)
  - **Non-target-trials** (test speaker ≠ model speaker)
- We run them through the system and record the scores
- We need to set the detection **threshold**



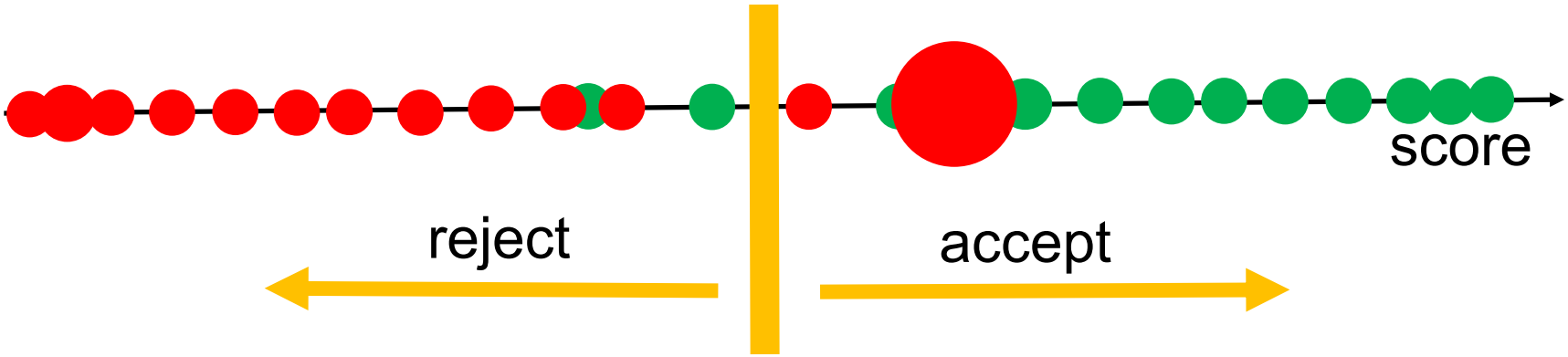
True accept



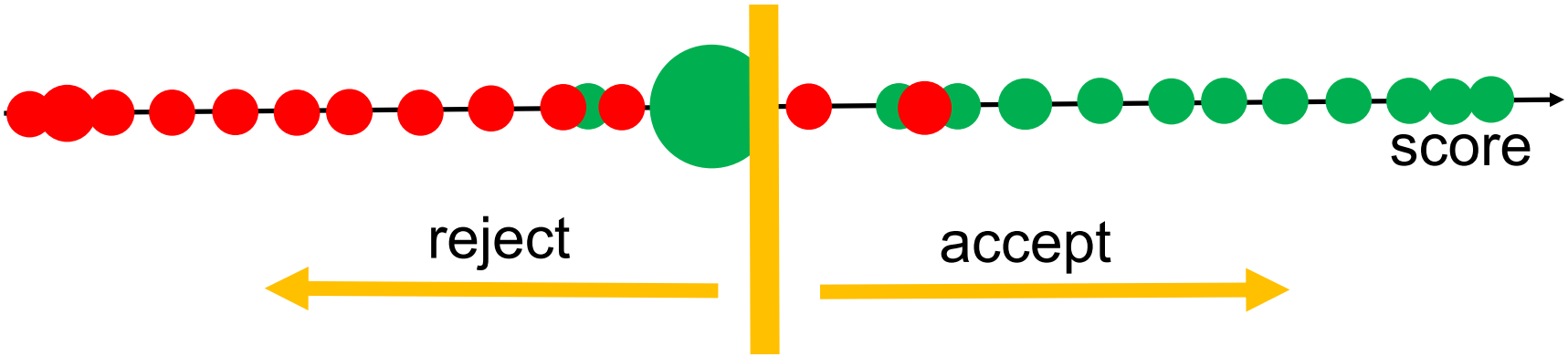
True reject



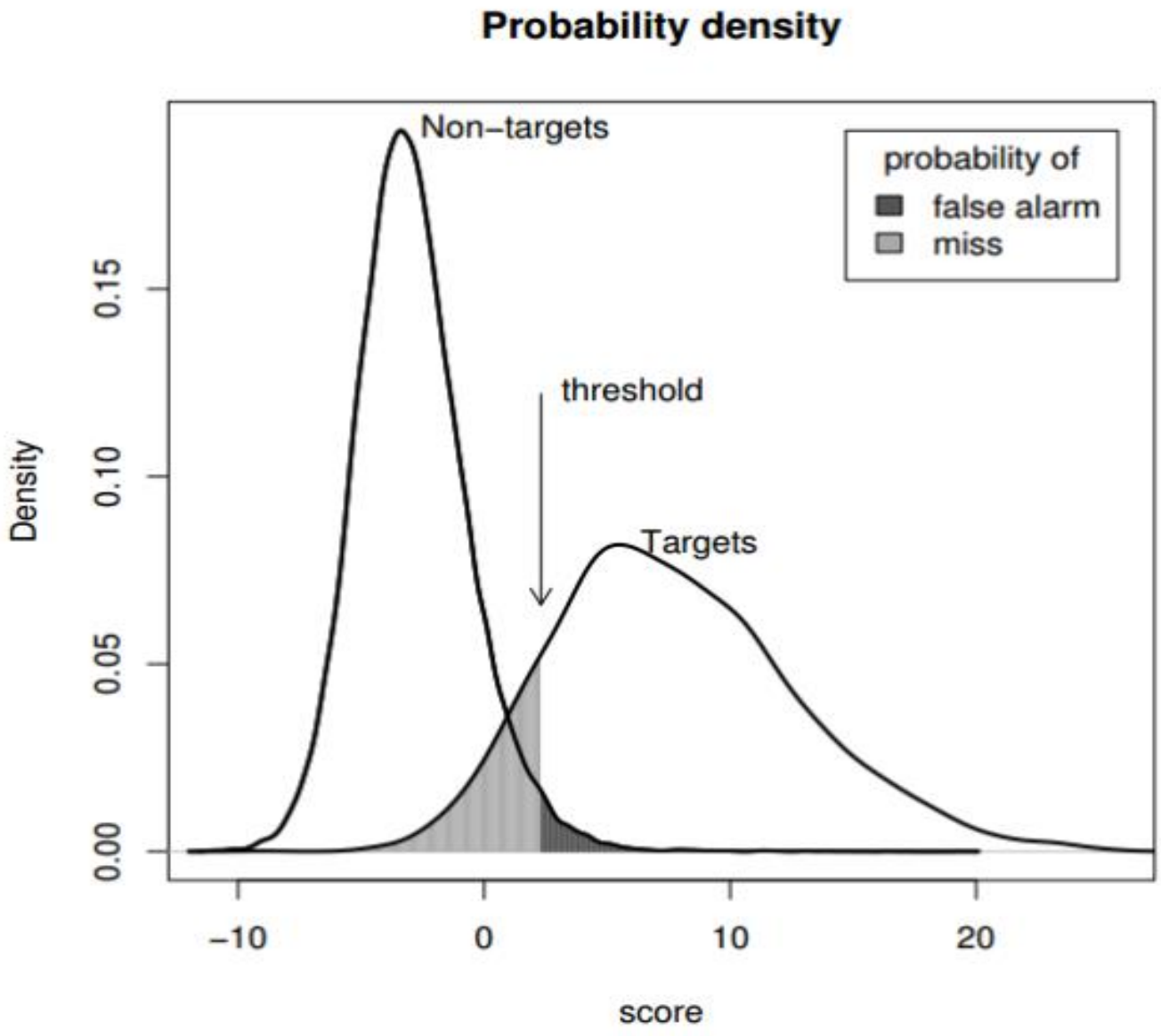
False alarm



Miss



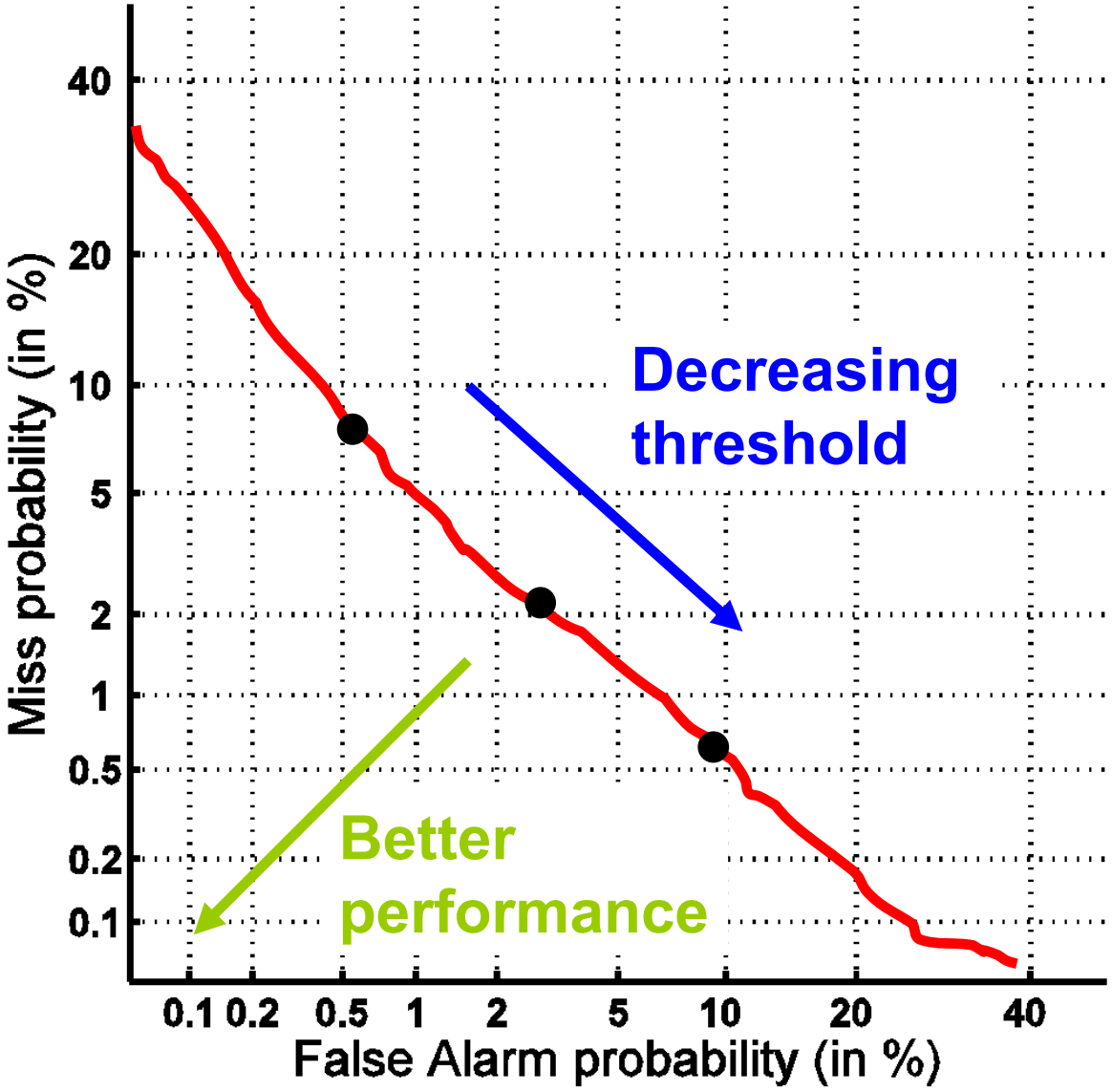
Single threshold, two types of error



False accept

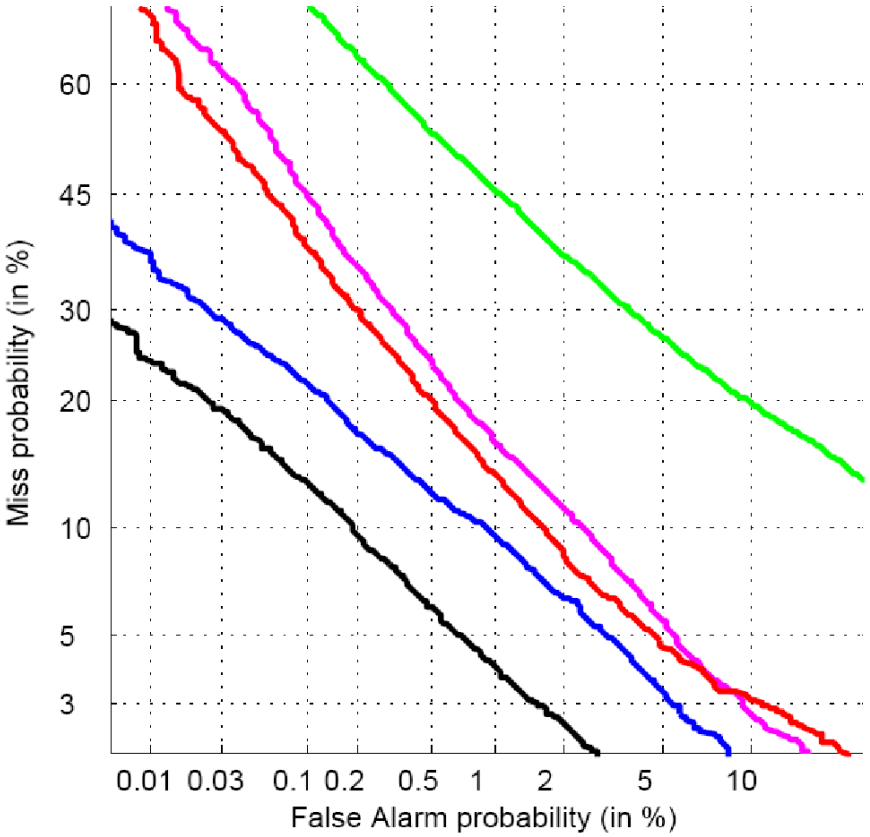
False reject

DET – Detection Error Tradeoff



Comparison of different SRE techniques

NIST SRE 2010, tel-tel (cond. 5)

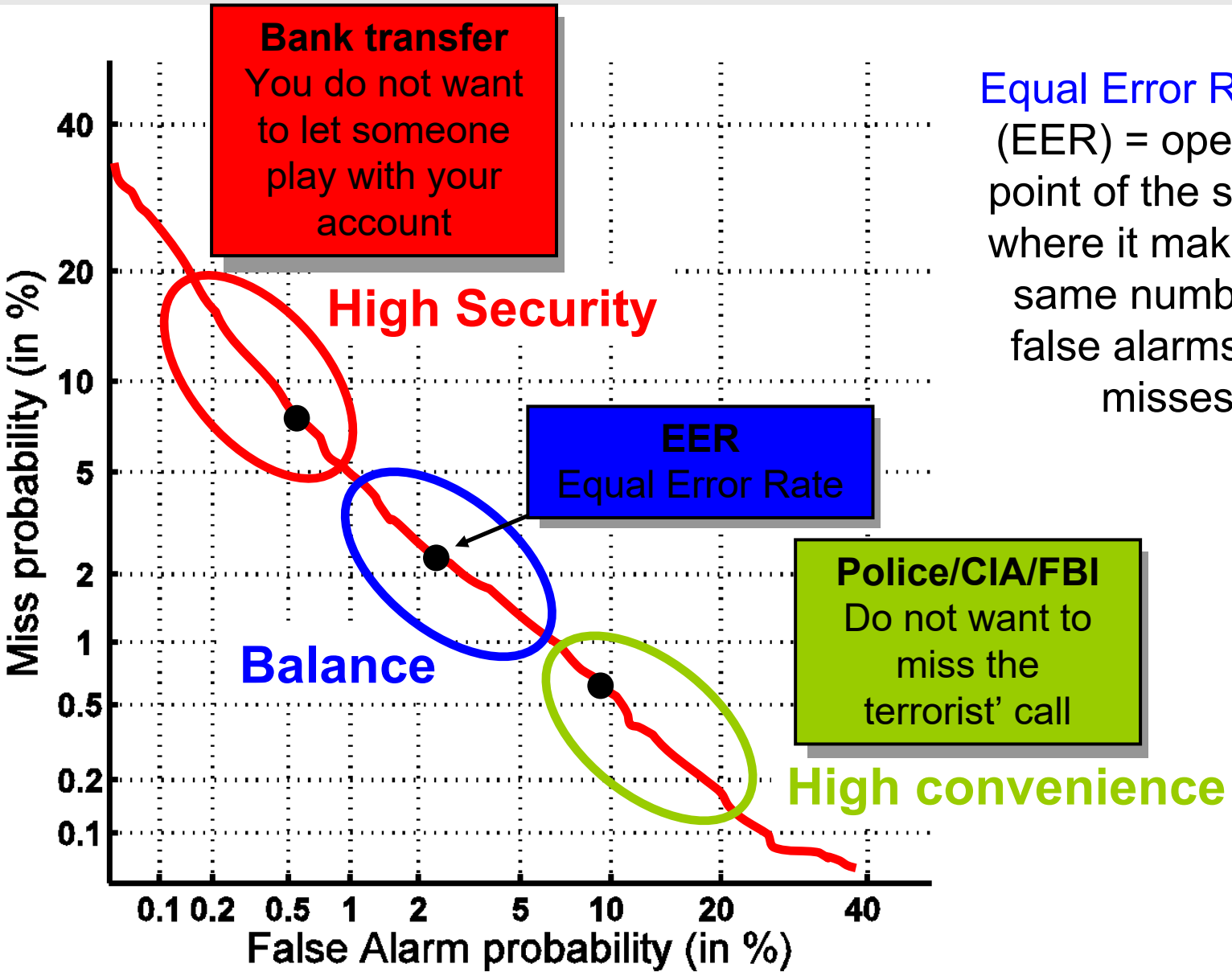


- Baseline (relevane MAP)
- Eigenchannel adapt.
- JFA
- iVector+PLDA
- iVector+PLDA - **current BUT system**

Additional significant recent improvements obtained with:

- full covariance UBM
- additional iVector pre-processing (LDA dim. reduction, length normalization,…)

DET – Detection Error Tradeoff



Equal Error Rate (EER) = operating point of the system where it makes the same number of false alarms and misses

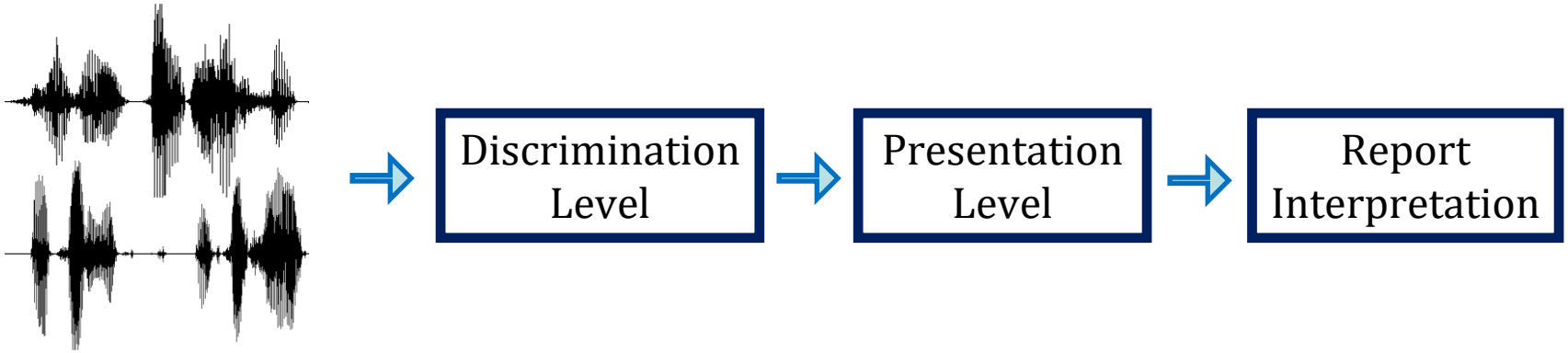
Agenda

- Speaker recognition applications
  - Why we need speaker recognition
  - Different application domains
- Background and theory
  - Feature extraction
  - Gaussian Mixture Modelling
  - Modelling of embeddings, PLDA
  - DNN embeddings
- Evaluation metrics
- Score calibration and normalization
- Conclusions

Score Calibration

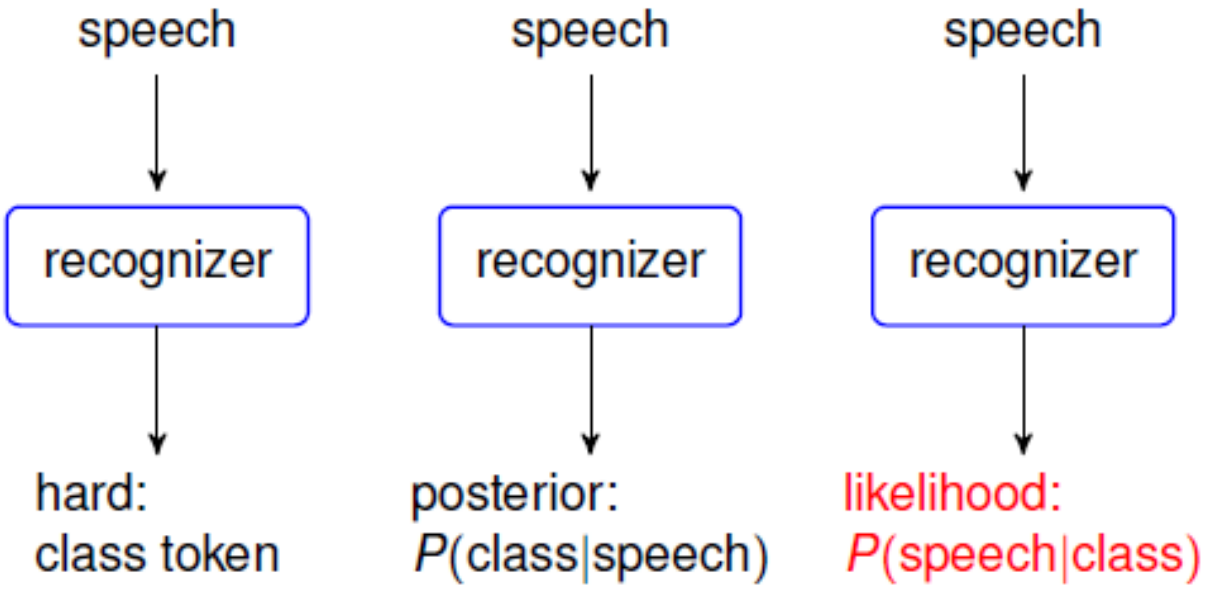
- We want a probabilistic interpretation of scores
  - Performing calibration often results in such scores
- Well calibrated scores have the same meaning on different sets and different systems – we can directly compare the numbers
- Calibration does not change system discriminability – it is usually only linear transformation
- Given the calibrated scores, we can set the threshold analytically for the desired operating point (application)

Score presentation



- Generally it is expected that systems produce scores (ideally log likelihood ratios):
  - >0 … more positive more sure it is the speaker
  - <0 … more negative more sure it is not the speaker

Scores/system outputs



posterior probability

likelihood

prior probability of class (here usually target trial)

$$P(\mathrm{w}|x) = \frac{P(x|\mathrm{w})P(\mathrm{w})}{P(x)}$$

Evidence

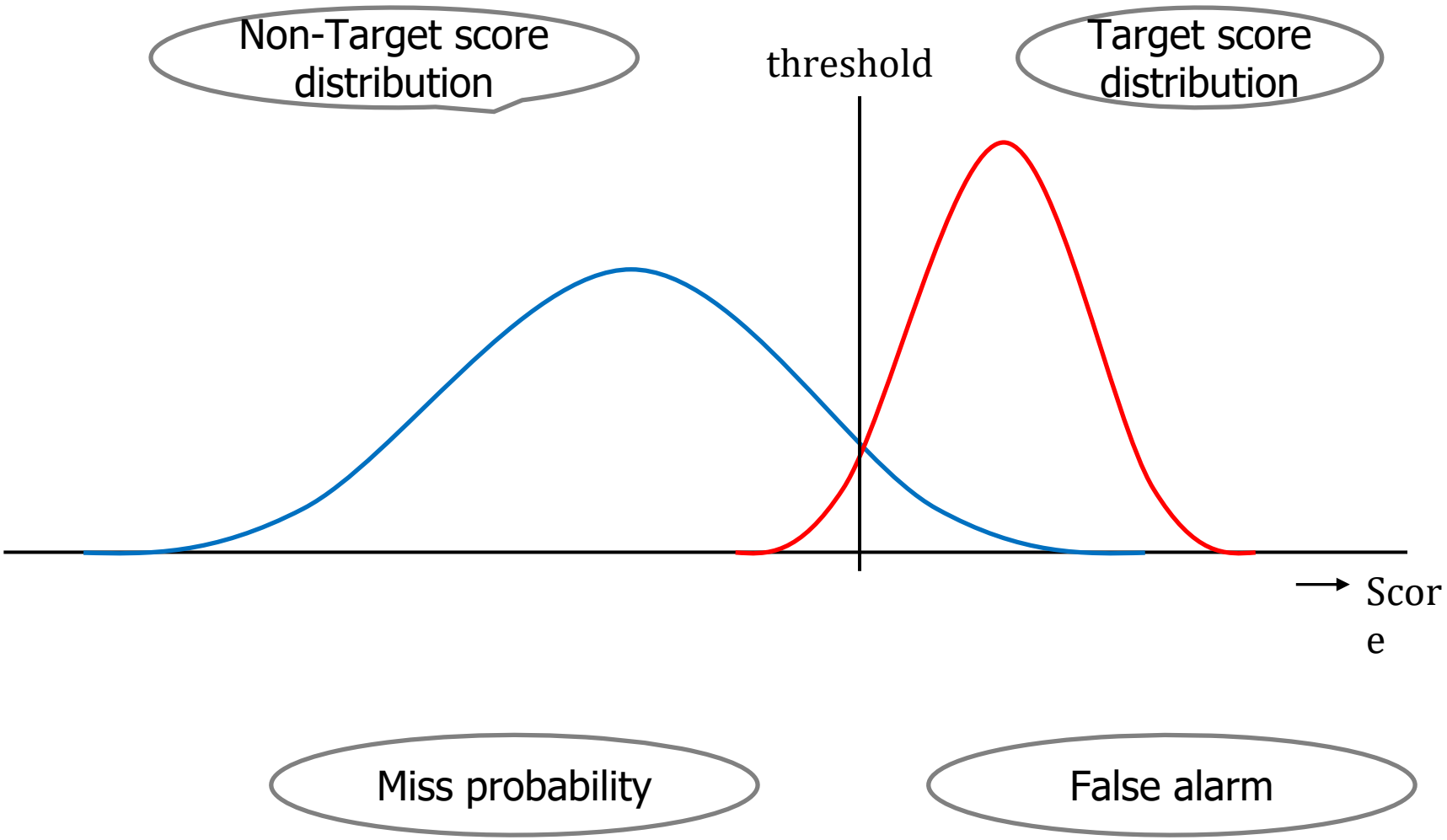
Scores

- The output of the recognizer is usually a score, which reflects the confidence of the system (ideally for SRE **log-likelihood ratio**).

$$s = \log \frac{p(t|H_s)}{p(t|H_d)}.$$

- Higher value reflects a higher confidence for same-speaker hypothesis and lower value for different-speaker hypothesis.
- Eventually, the score is converted to a hard decision by thresholding.
- Moving the threshold changes the proportion of the two error rates (pmiss, pfa)

Discriminability of the system



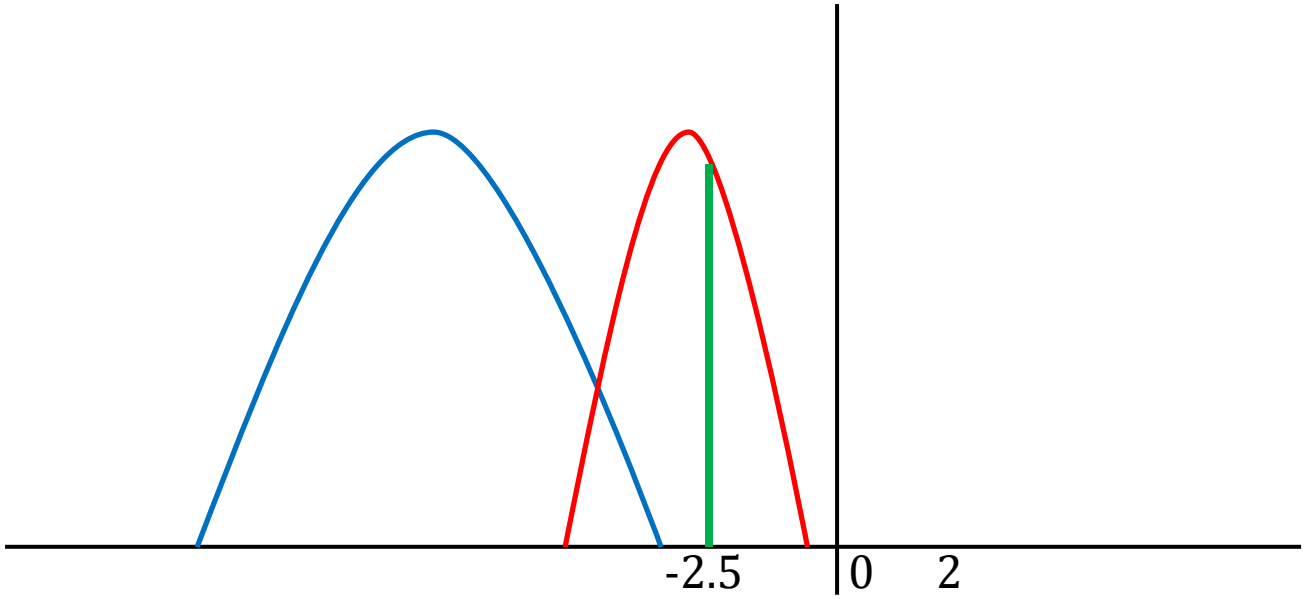
Calibrating the scores

- Provided scores are often not very nice LLR
- Calibration task is to transform the scores in such a way that we get LLR and therefore we can set ideal threshold (for given operating point)
- Problems often arise from the dataset shift (typically development data vs target, evaluation data)
- Usually a simple transformation f(s) (monothonic increasing) is enough to calibrate:

$$f(s) = a\,s + b,$$

  - Parameters a and b can be found by optimizing a cross entropy objective over a supervised development set by the means of logistic regression – **discriminative approach**

Calibration



- Miss-calibrated system because of different channel, language, noise, reverberation ….

Detection Cost Function (DCF)

- Defined by NIST as a metric for evaluating the verification systems, which focuses on a particular operation point of interest.
- It is designed to consider the overall costs based on the two types of detection errors.

$$\mathrm{DCF} = C_{\mathrm{miss}}\, p(\mathrm{miss}|\mathcal{T}, \tau)\, p(H_s) + C_{\mathrm{fa}}\, p(\mathrm{fa}|\mathcal{T}, \tau)\, p(H_d)$$

$$p(H_d) = 1 - p(H_s)$$

triplet $\langle C_{\mathrm{miss}}, C_{\mathrm{fa}}, p(H_s)\rangle$ defines the target *operating point*

	C_{fa}	C_{miss}	$p(H_s)$
$\mathrm{DCF}_{\mathrm{old}}$	1	10	0.01
$\mathrm{DCF}_{\mathrm{new}}$	1	1	0.001

Setting for NIST SRE2008 and 2010

Measuring calibration loss

- DCF (actual) is computed from the actual hard decisions.
- Threshold for making decisions is often set by evaluator
- NIST also computes a minimum possible DCF

$$\min \mathrm{DCF} = \min_{\tau} \; [C_{\mathrm{miss}}\, p(\mathrm{miss}|\mathcal{T},\tau)\, p(H_s) + C_{\mathrm{fa}}\, p(\mathrm{fa}|\mathcal{T},\tau)\, p(H_d)] \,.$$

- The difference between the act-DCF and min-DCF is referred to as a calibration loss.
- Calibration task is to transform the scores in such a way that we get least errors at the given operating point

Analytically setting the threshold

- If the scores are well-calibrated LLR, user can set the threshold analytically to make an optimal, cost-ffective Bayes decision

$\langle C_{\text{miss}}, C_{\text{fa}}, p(H_s) \rangle$ can be absorbed to a single *effective prior* P_{tar}

$$\text{logit}\,\text{P}_{\text{tar}} = \text{logit}(p(H_s)) + \log\frac{C_{\text{miss}}}{C_{\text{fa}}}. \qquad \text{logit}(x) = \log\frac{x}{1-x}$$

Analytical and optimal threshold for the scores (LLRs) is then $\tau = -\text{logit}\,\text{P}_{\text{tar}}$.

Conclusions

- Over the past decade
  - error rates have decreased 5 times or more thanks to advances in channel compensation techniques
  - the new techniques have resulted in massive speed-ups
    - i-vector/x-vectors are extracted for each recording (about 100 times faster than RT)
    - With embeddings, **billions** of verification trials can be tested in few seconds
- DNN embeddings have replaced generative embeddings (i-vectors)
- Use of large SSL models is promising when little annotated training data are available for training DNN embedding extractors
- Channel distortion is still the big issues for correct recognition
  - Calibration and score normalization is necessary for practical systems
- **For more details talk to anybody from Speech@FIT**

Thanks for your attention
and
I hope you enjoyed it ;)