ECE277/477

A Brief Introduction to Speaker modeling in Speaker Recognition

Ge Zhu

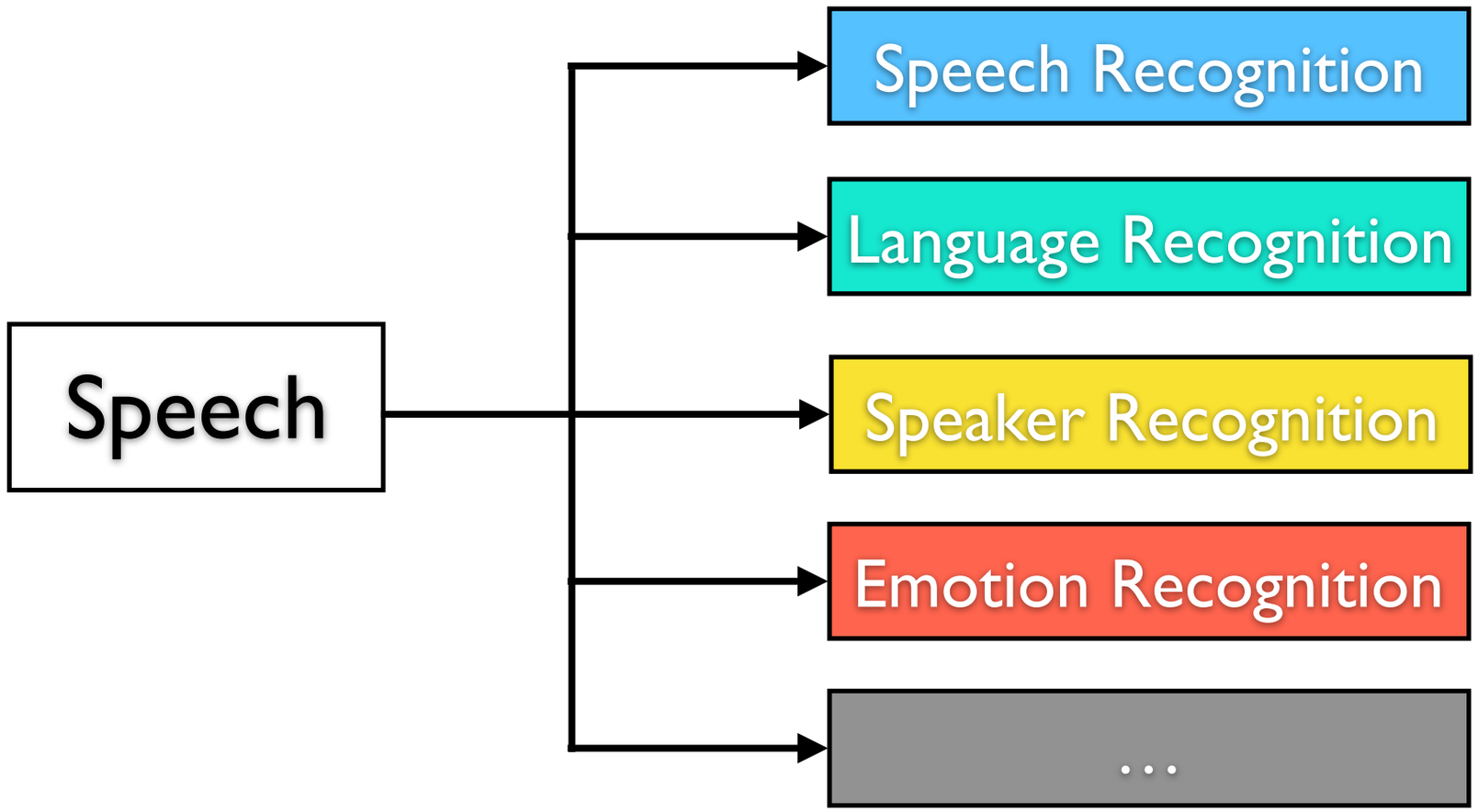
Outline

- Background and Introduction
- Part I: Traditional Models
  - GMM-UBM
  - Factor Analysis, i-vector
- Part II: Deep Models
  - Deep Representation Learning
  - Metric Learning

Background: Speech as HCI

- Goal:

  Extract Information transmitted in speech signal



Introduction: Speaker Recognition

- Speaker Verification:

  Supervised binary classification: Given a speech sequence and a claimed identity, accept or reject the identity.

- Speaker Identification:

  Supervised multi-class classification: Determine which speaker (from a predetermined set of speakers) has uttered the sequence.

- Speaker Diarization:

  Clustering and segmentation: Partition an input audio stream into homogeneous segments. according to the

Feature Learning Speaker Verification System

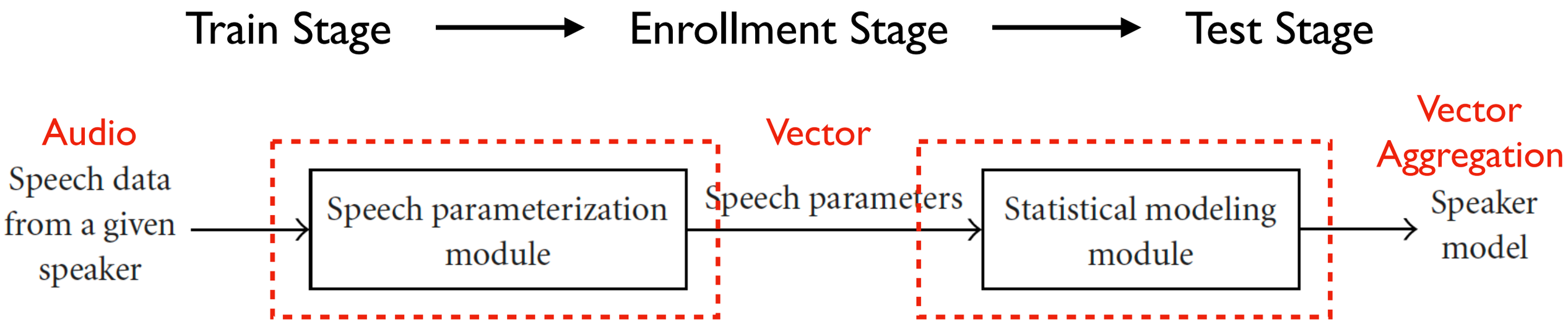


FIGURE 1: Modular representation of the training phase of a speaker verification system.

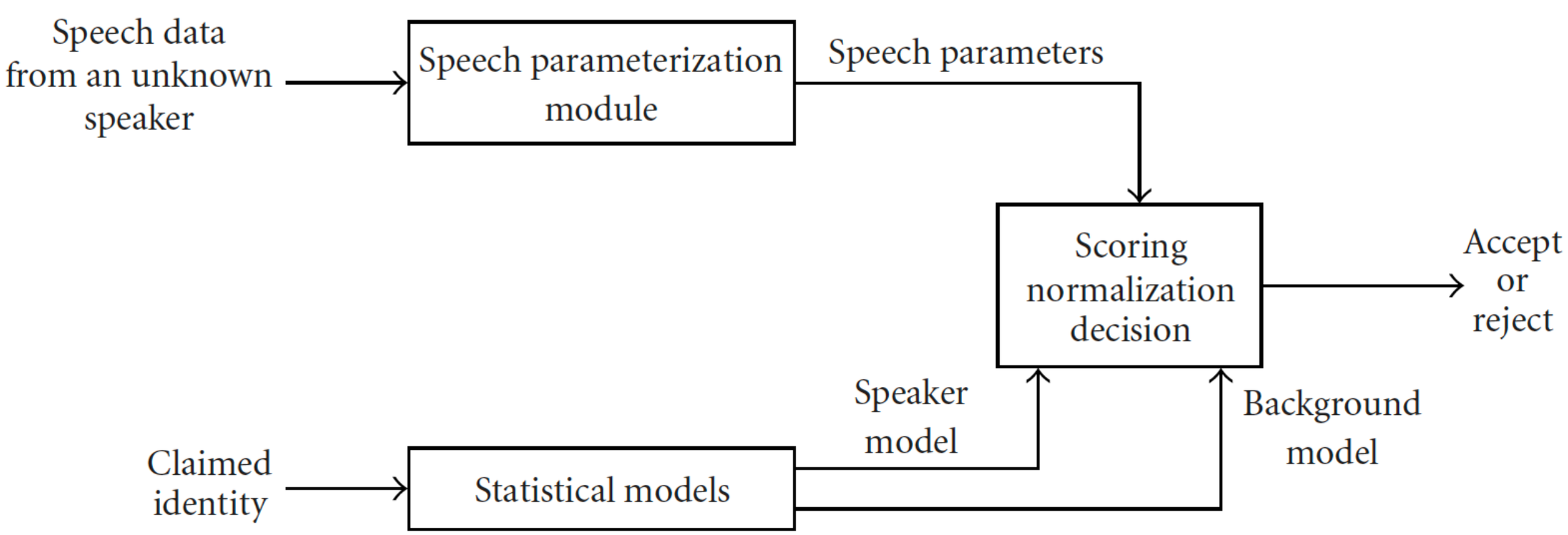
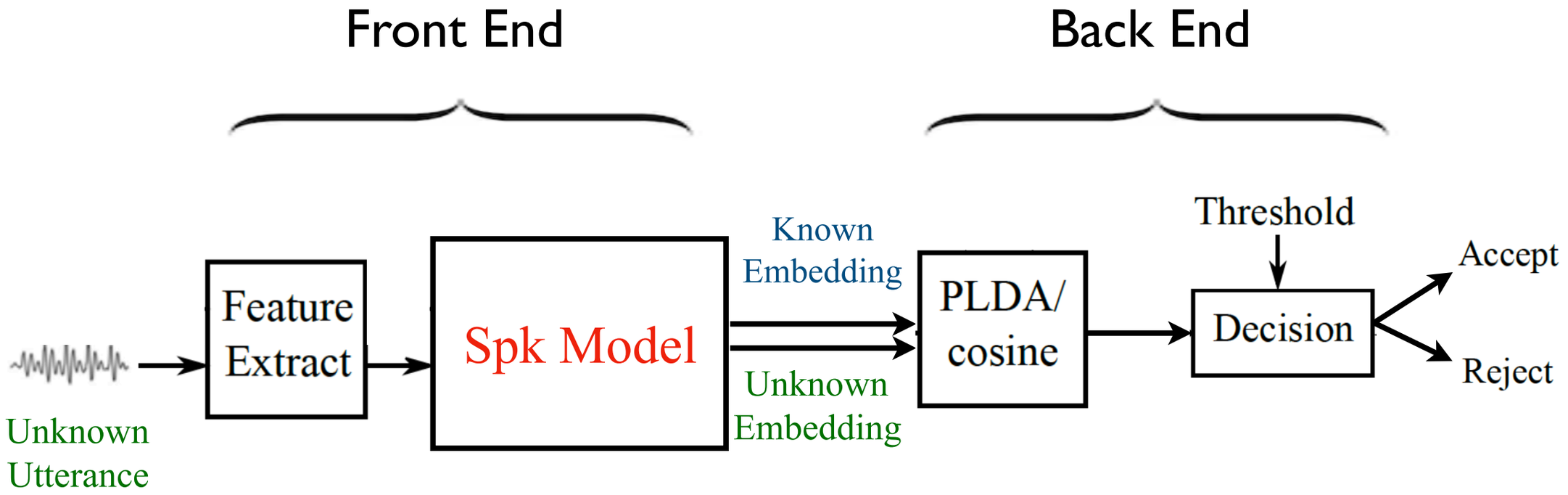


FIGURE 2: Modular representation of the test phase of a speaker verification system.

In Detail:


Front End
Back End
Unknown Utterance
Feature Extract
Spk Model
Known Embedding
Unknown Embedding
PLDA/ cosine
Threshold
Decision
Accept
Reject

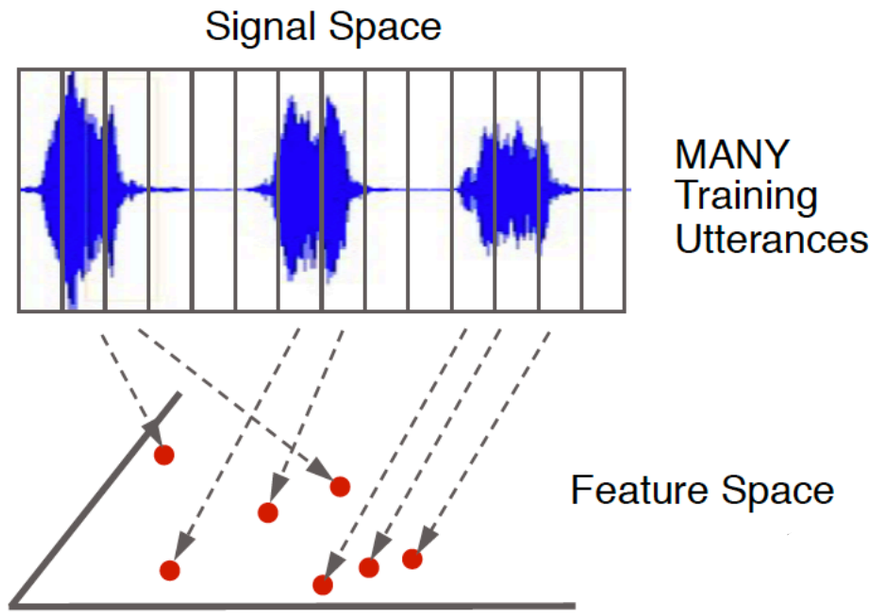
Feature Extraction

- Spectral Features:

  Spectrogram

- Cepstral Features:

  MFCC, Melspectrum…



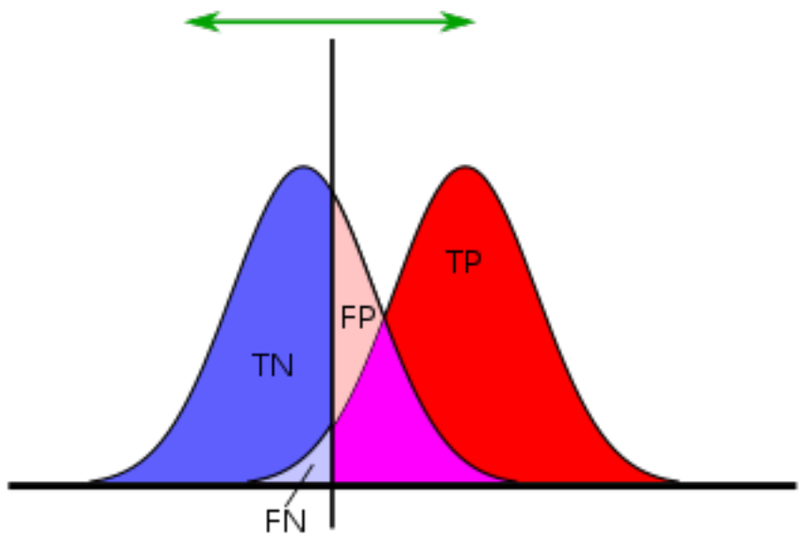
Evaluate Metrics

- EER (Equal Error Rate):

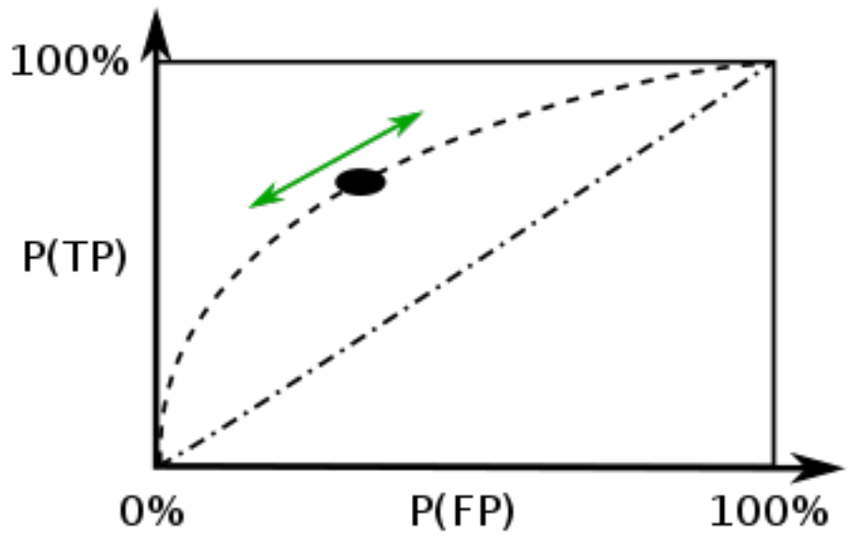
  In evaluate stage, pairs of known identity (enrolled utterance) and unknown utterance is tested, equal error rate refers to the value when false positive rate = false negative rate.

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

Evaluate Metrics



TP	FP
FN	TN



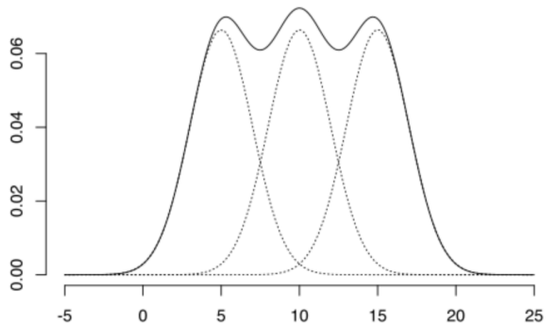
Speaker Model: Problem formulation

Representation Learning Problem:

- Build a speaker dependent model based on frequency domain features, similar to timbre representation; (Hopefully independent to other factors.)

Part I:
Traditional Method: GMM

- Speech is produced by vocal-tract configuration, we can only observe timbre/spectral features, similar to HMM



Gaussian Mixture Models

- Use Gaussian to model state emission probability:

$$\mathcal{N}(x \mid \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_i \mid^{1/2}} exp\{-\frac{1}{2}(x-\mu_i)^T(\Sigma_i)^{-1}(x-\mu_i)\}$$

μi represents expected spectral feature vector from state i

Σi represents correlations and variability of spectral features within state i

GMM for speaker model

- Input:

Sequence feature X, $X=\{x_1,x_2,\ldots x_T\}$

- M-state GMM model, for a vector feature x_t

$$p(\boldsymbol{x_t}\,|\,\lambda) = \sum_{i=1}^{M} \pi_i \mathcal{N}(\boldsymbol{x_t}\,|\,\mu_i, \Sigma_i), \text{ with } \sum_{i=1}^{M} \pi_i = 1$$

parameter: $\lambda = (p_i, \mu_i, \Sigma_i)$, for i = 1, 2, …, M

- Probability for a sequence:

$$\log p(X\,|\,\lambda) = \log \prod_{t=1}^{T} p(\boldsymbol{x_t}\,|\,\lambda) = \sum_{t=1}^{T} \log(\sum_{j=1}^{M} p_j b_j(\boldsymbol{x_t}))$$

$$\text{Posterior probability} \propto \text{Likelihood} \times \text{Prior probability}$$

$$p(\theta|x) = \frac{p(x|\theta)}{p(x)} p(\theta)$$

GMM Speaker Model Parameter Estimation

- Input: features from speaker s
- Objective function (MLE):

$$\lambda_S = \arg\max_{\lambda} \log p(X \mid \lambda)$$

- Algorithm: Expectation Maximization

Solving GMM with EM

- Define LL(log likelihood):

$$LL(\lambda) = \log p(X \mid \lambda) = \sum_{t}^{T} \log p(\boldsymbol{x}_t \mid \lambda)$$

- Optimize:

$$\max LL(\lambda)$$

$$s.t. \sum_{i=1}^{M} \pi_i = 1$$

- Hard to optimize:

(a)no closed form solution; (b)constraint on π.

Solving GMM with EM

Iterate:
E: Construct a bound
M: Optimize the bound

$$p(\boldsymbol{x_t} \mid \lambda) = \sum_{i=1}^{M} \pi_i \mathcal{N}(\boldsymbol{x_t} \mid \mu_i, \Sigma_i) \qquad (p(\boldsymbol{x_t} \mid \lambda) = \sum_{i=1}^{M} p(z_i \mid \lambda) p(\boldsymbol{x_t} \mid z_i, \lambda))$$

Here π_i can be seen as a latent variable z distribution p(z|λ) ,therefore:

$$LL(\lambda) = \sum_{t}^{T} \sum_{z_t} \log p(\boldsymbol{x_t}, z_t \mid \lambda)$$

Jensen's Inequality: if Φ is convex, then $\phi(\boldsymbol{E}(x)) \leq \boldsymbol{E}(\phi(x))$.

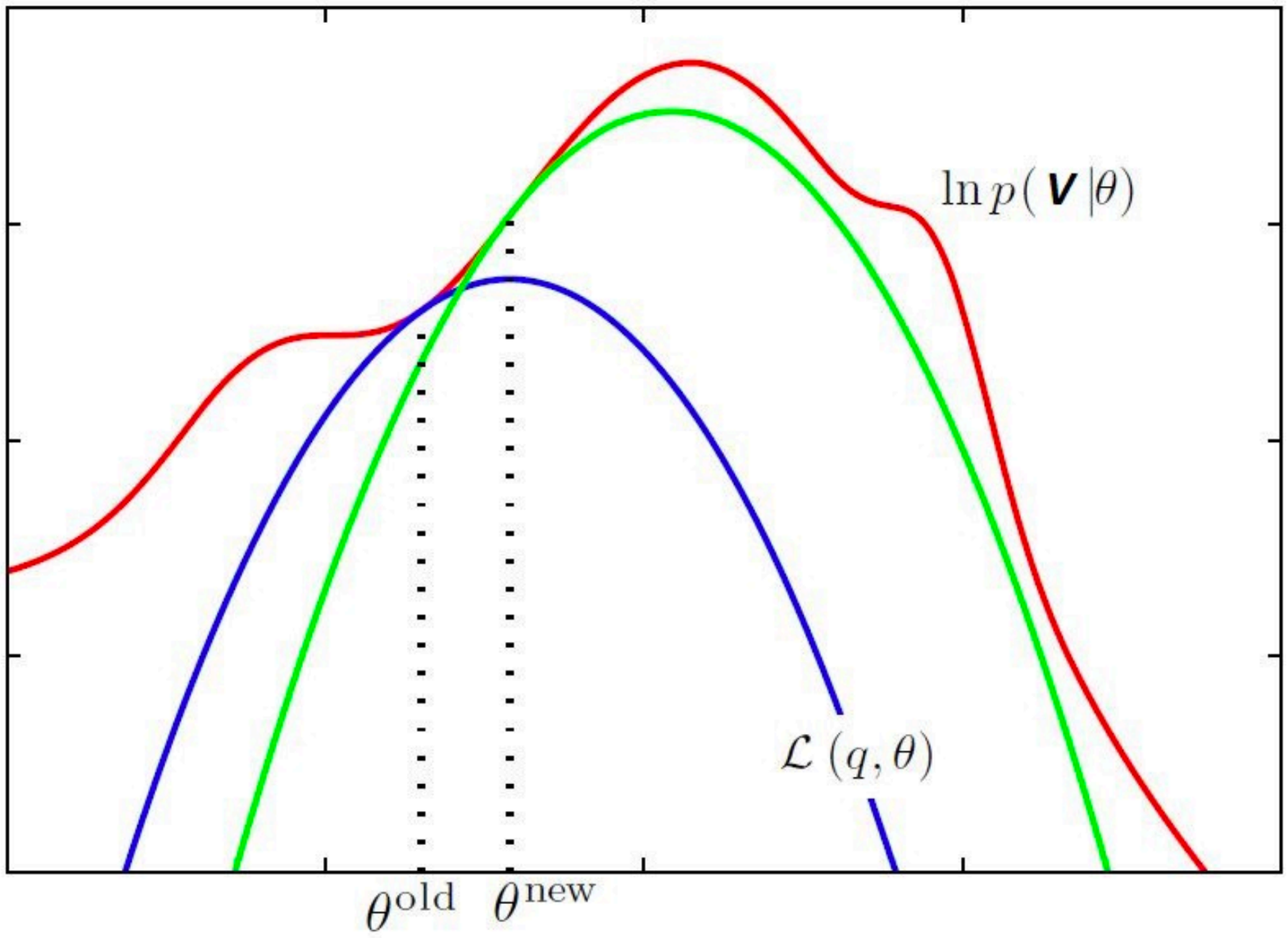
$$LL(\lambda) = \sum_{t}^{T} \sum_{z_t} \log[p(z_t \mid \boldsymbol{x_t}, \lambda') \frac{p(\boldsymbol{x_t}, z_t \mid \lambda)}{p(z_t \mid \boldsymbol{x_t}, \lambda')}]$$

← Current Parameter

← Previous Parameter

$$\geq \sum_{t}^{T} \sum_{z_t} p(z_t \mid \boldsymbol{x_t}, \lambda') \log[\frac{p(\boldsymbol{x_t}, z_t \mid \lambda)}{p(z_t \mid \boldsymbol{x_t}, \lambda')}] = \sum_{t}^{T} \boldsymbol{E}_{z_t} \log[\frac{p(\boldsymbol{x_t}, z_t \mid \lambda)}{p(z_t \mid \boldsymbol{x_t}, \lambda')}] = Q(\lambda, \lambda')$$

Expectation Maximization:



Solving GMM with EM

- Therefore we construct a lower bound function:

$$LL(\lambda) \geq Q(\lambda, \lambda')$$

- Optimize

$$\lambda = \arg\max_{\lambda} Q(\lambda, \lambda')$$

- Therefore:

$$LL(\lambda) \geq Q(\lambda, \lambda') \geq Q(\lambda', \lambda') \geq LL(\lambda')$$

Gibbs's Inequality: $-\sum_{i=1}^{n} p_i \log p_i \leq -\sum_{i=1}^{n} p_i \log q_i$

*For implementation details and derivations: (reference from PRML)

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

where

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk}). \tag{9.27}$$

4. Evaluate the log likelihood

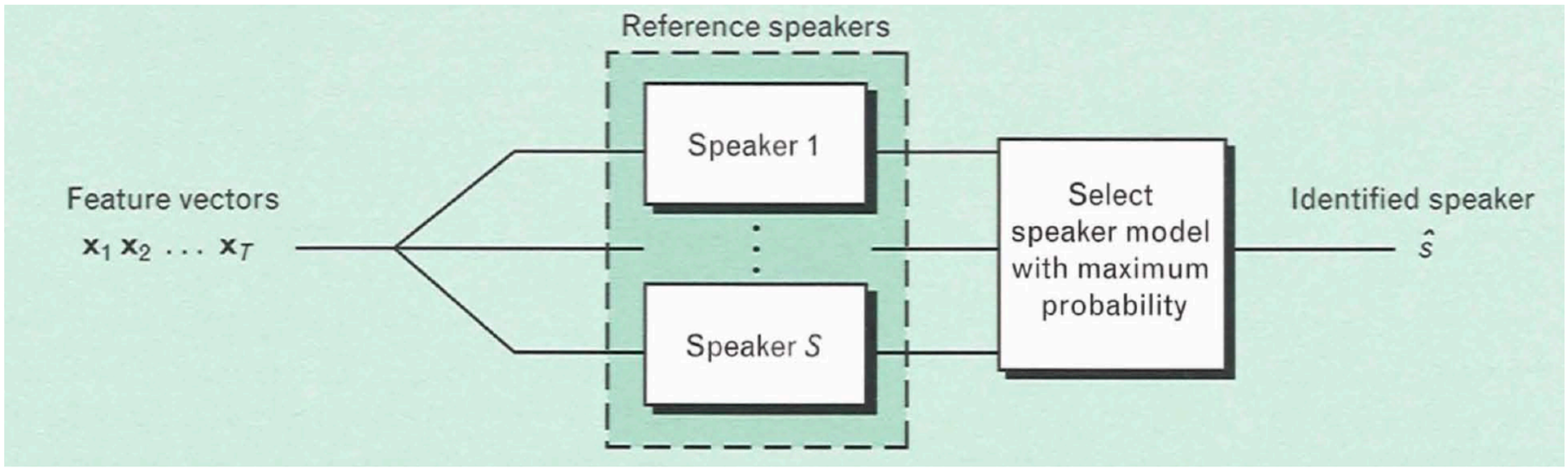
$$\ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right\} \tag{9.28}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

So far…

- We extract speaker parameters by using M-state Gaussian parameters, we can calculate probability for an utterance:

$$\log p(X|\lambda) = \log \prod_{t=1}^{T} p(\boldsymbol{x}_t|\lambda) = \sum_{t=1}^{T} \log\Big(\sum_{j=1}^{M} p_j b_j(\boldsymbol{x}_t)\Big)$$

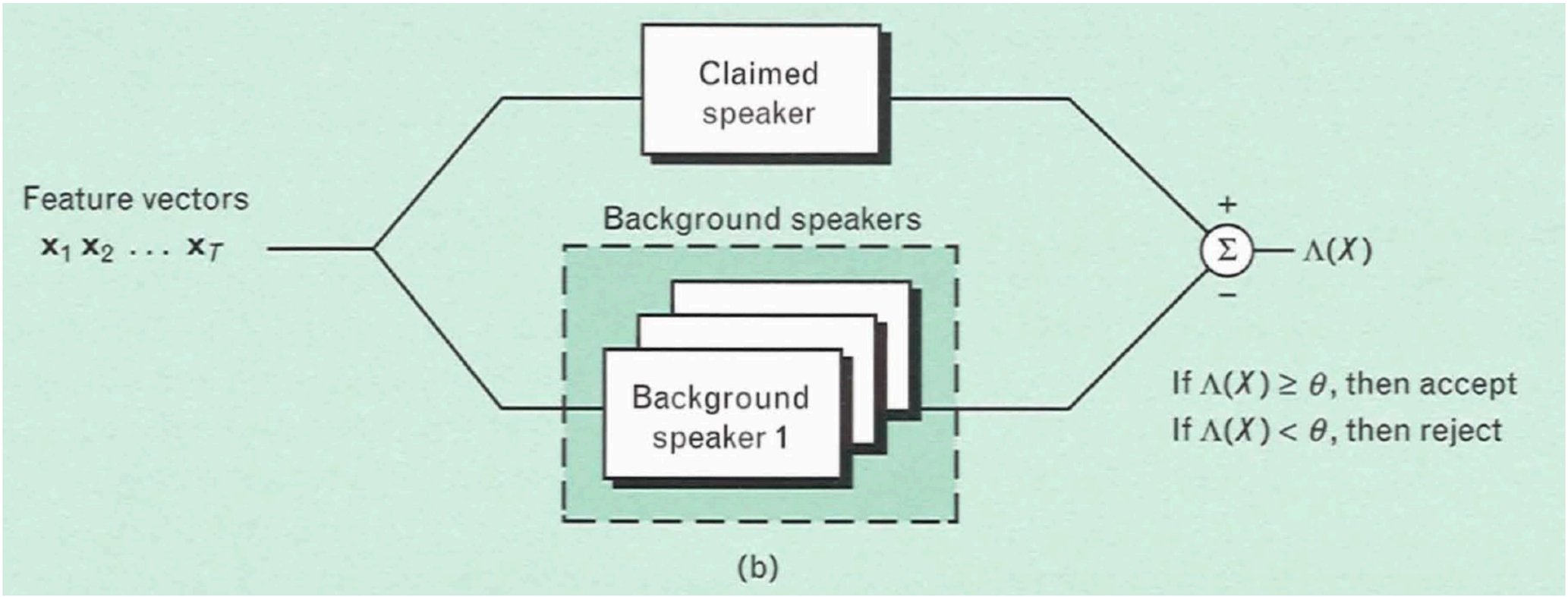


Speaker Identification

Background Model

- In GMM model, decision is made by comparing likelihood: speaker identification
- Require an alternate speaker model (nonclaimed speaker model) for speaker verification

Speaker Verification:



(b)

Background Model

- LR (Likelihood Ratio) test:

$$\frac{Pr(X \text{ is from the claimed speaker})}{Pr(X \text{ is not from the claimed speaker})} = \frac{Pr(\lambda_C|X)}{Pr(\lambda_{\bar{C}}|X)} = \frac{Pr(\lambda_C)Pr(X|\lambda_C)}{Pr(\lambda_{\bar{C}})Pr(X|\lambda_{\bar{C}})} \lessgtr C$$

$$\Lambda(X) = \frac{Pr(X|\lambda_C)}{Pr(X|\lambda_{\bar{C}})} \lessgtr \frac{Pr(\lambda_{\bar{C}})}{Pr(\lambda_C)} = \Gamma$$

- Compare LR with prior ratio threshold

Solution

Selection problem

- 1. A model composed of N background speaker models for one speaker:

$$Pr(X|\lambda_{\bar{C}}) = f(Pr(X|\lambda_1), \ldots, Pr(X|\lambda_N))$$

- 2. Train a single model composed of speech from several speakers.

Universal Background Model

- UBM: a large GMM to represent speaker independent distribution of features
- Maximum a-posterior (MAP) estimate:

$$\lambda_S = \arg\max_{\lambda} \log p(X \mid \lambda) p(\lambda)$$

- Intuition: use priori knowledge over data domain when training speaker dependent data
- Parameter Estimation: *Conjugate Prior

Summary on GMM-UBM

- Pros

  1. Couple SD model with SI together, more robust;
  2. Less data is required for extracting SD features;
  3. Easy for scoring for Speaker verification system.

- Cons

  1. High dimension GMM;
  2. Sensitive to acoustic environment.

Traditional Method: i-vector

- Representation for an utterance of unfixed length: GMM supervector (high dimension);
- GMM supervector: A large vector concatenating the parameters of GMM model, such as GMM mean vectors.

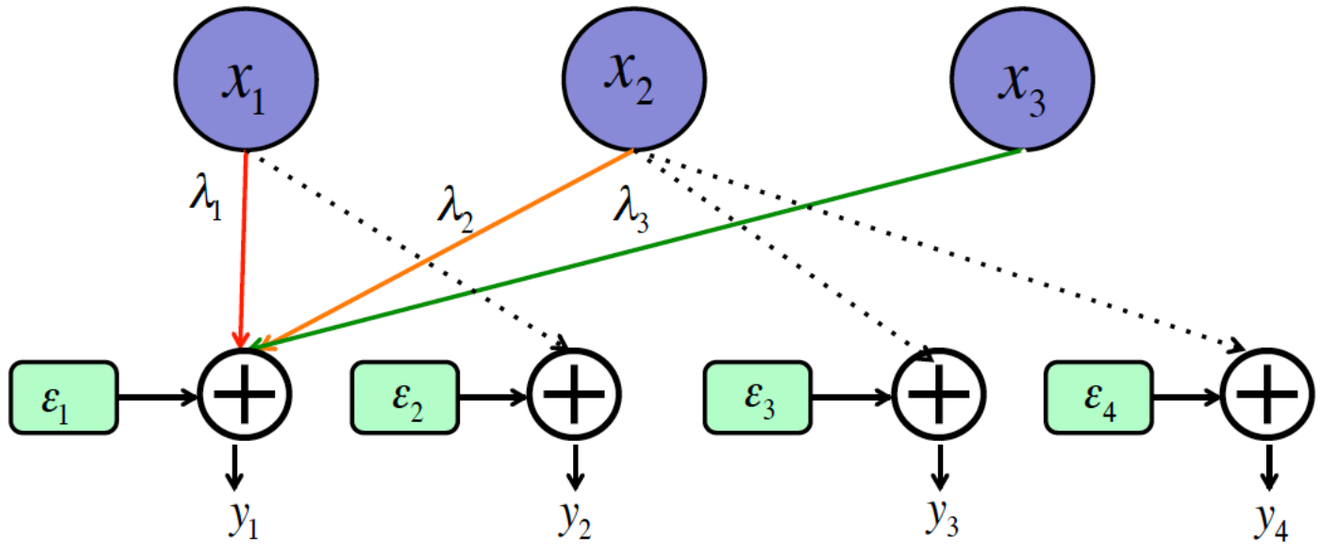
Factor Analysis

- In real problem: separate acoustic environment factors from speaker identity factor

In GMM-UBM, we try to couple dataset domain knowledge (SI factor) as prior knowledge into SD parameter estimation.

Factor Analysis

- Dimension Reduction: modeling variability in high-dimension observable data vectors with lower hidden variables;



FA: Similar architecture to NN, but no activations

Factor Analysis

- Form:

$$y - \mu = \Lambda x + \epsilon$$

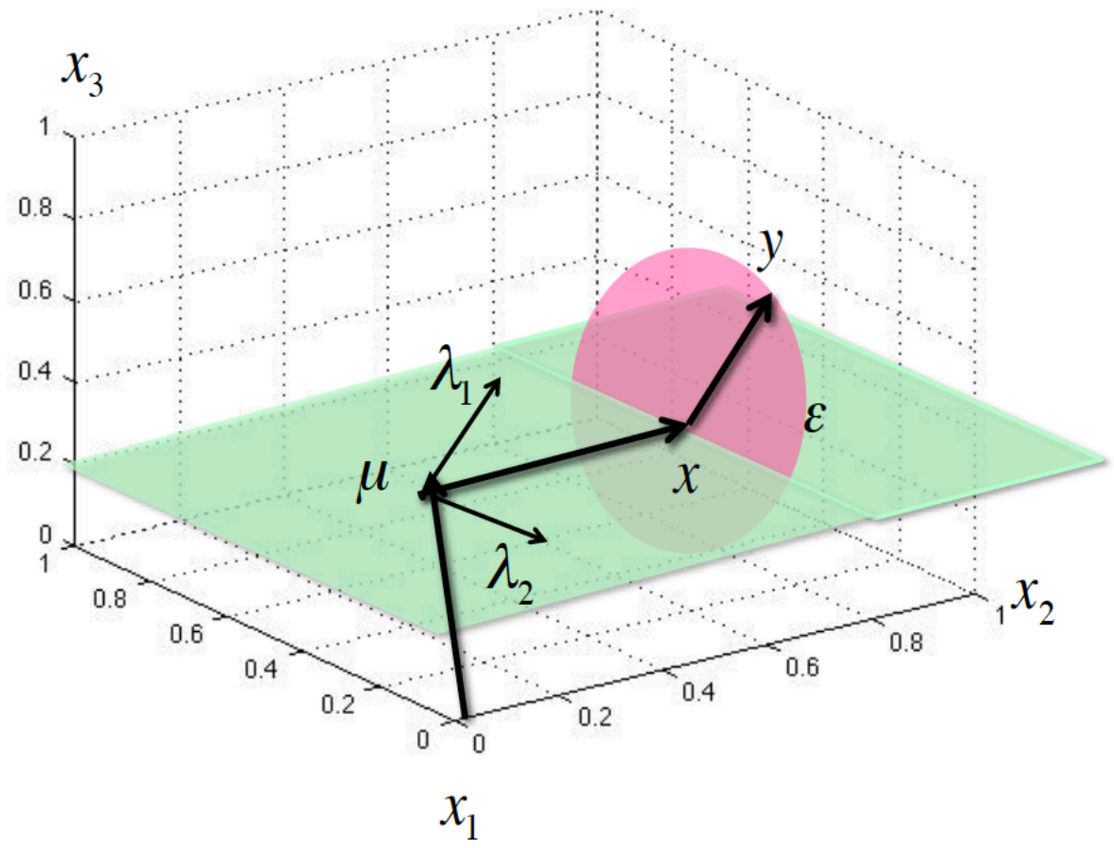
y: data vector

μ: mean vector

Λ: loading matrix

x: factor vector

ε: error vector



Factor Analysis

- Assumptions:

  1. x and ε are independent;

  2. E(x) = E(ε) = 0;

  3. Cov(x) = I, factors are independent.

Joint Factor Analysis

- In general, speaker model is decomposed as several components:

$$\mathbf{m}_{s,h} = \mathbf{m}_0 + \mathbf{m}_{\text{spk}} + \mathbf{m}_{\text{chn}} + \mathbf{m}_{\text{res}}$$

Independent

SPK dependent

Channel/Environment dependent

Noise

- Joint Factor Analysis (still high dimension):

$$\mathbf{m}_{s,h} = \mathbf{m}_0 + \mathbf{U}\mathbf{x}_h + \mathbf{V}\mathbf{y}_s + \mathbf{D}\mathbf{z}_{s,h}$$

i-Vector

- In fact, channel information still contain speaker information in JFA model!
- Modify JFA into total variability space model:

$$\mathbf{m}_{s,h} = \mathbf{m}_0 + \mathbf{T}\mathbf{w}_{s,h}$$

Speaker and Channel/Environment dependent

-T is total variability matrix

-m_0 is residual component

-$w_{s,h}$ is i(dentity)-vector

Summary on i-vector

- Pros

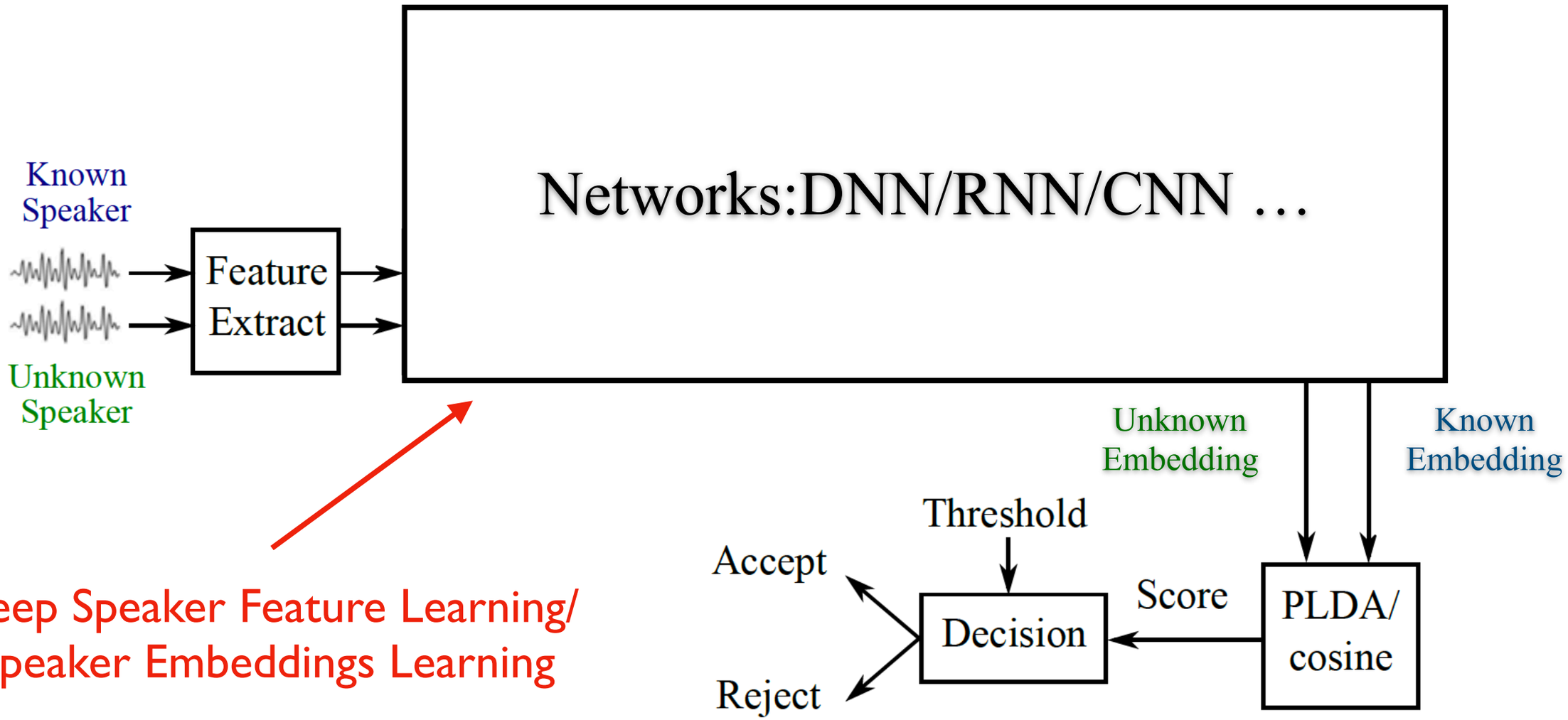
  Lower dimension, but keep information of GMM supervector, more applicable in compensation and scoring.

- (Possible) Cons

  GMM based feature extraction method.

Novel

SV System summary



Deep Speaker Feature Learning/
Speaker Embeddings Learning

Deep Model: d-Vector

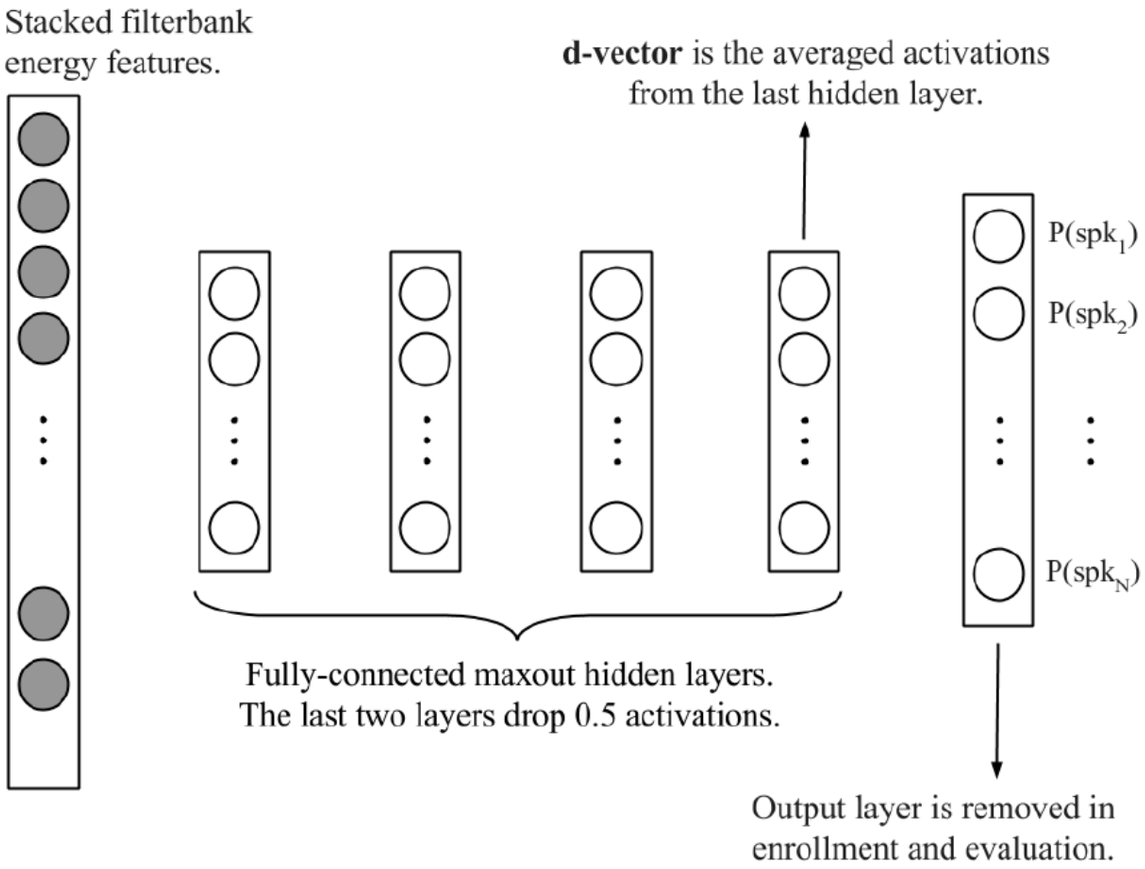
- Network: DNN
- Input: contextual MFCC $(\mathbf{x}_{k1}, \ldots, \mathbf{x}_{kM})$
- Output: L2 normalized neural net weights as embeddings $(\mathbf{e}_{k1}, \ldots, \mathbf{e}_{kM})$
- Enrollment: embeddings centroid

$$\mathbf{c}_k = \mathbb{E}_m[\mathbf{e}_{km}] = \frac{1}{M}\sum_{m=1}^{M}\mathbf{e}_{km}$$

- Evaluation: Cosine similarity Score $\cos(\mathbf{e}_{j\sim}, \mathbf{c}_k)$

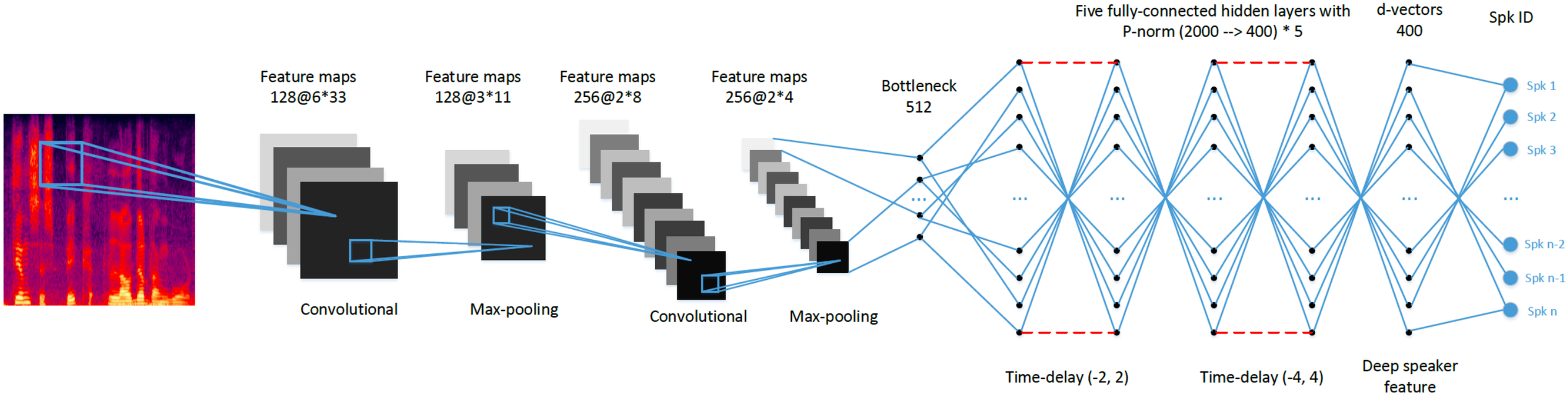
d-Vector

- Model Training:

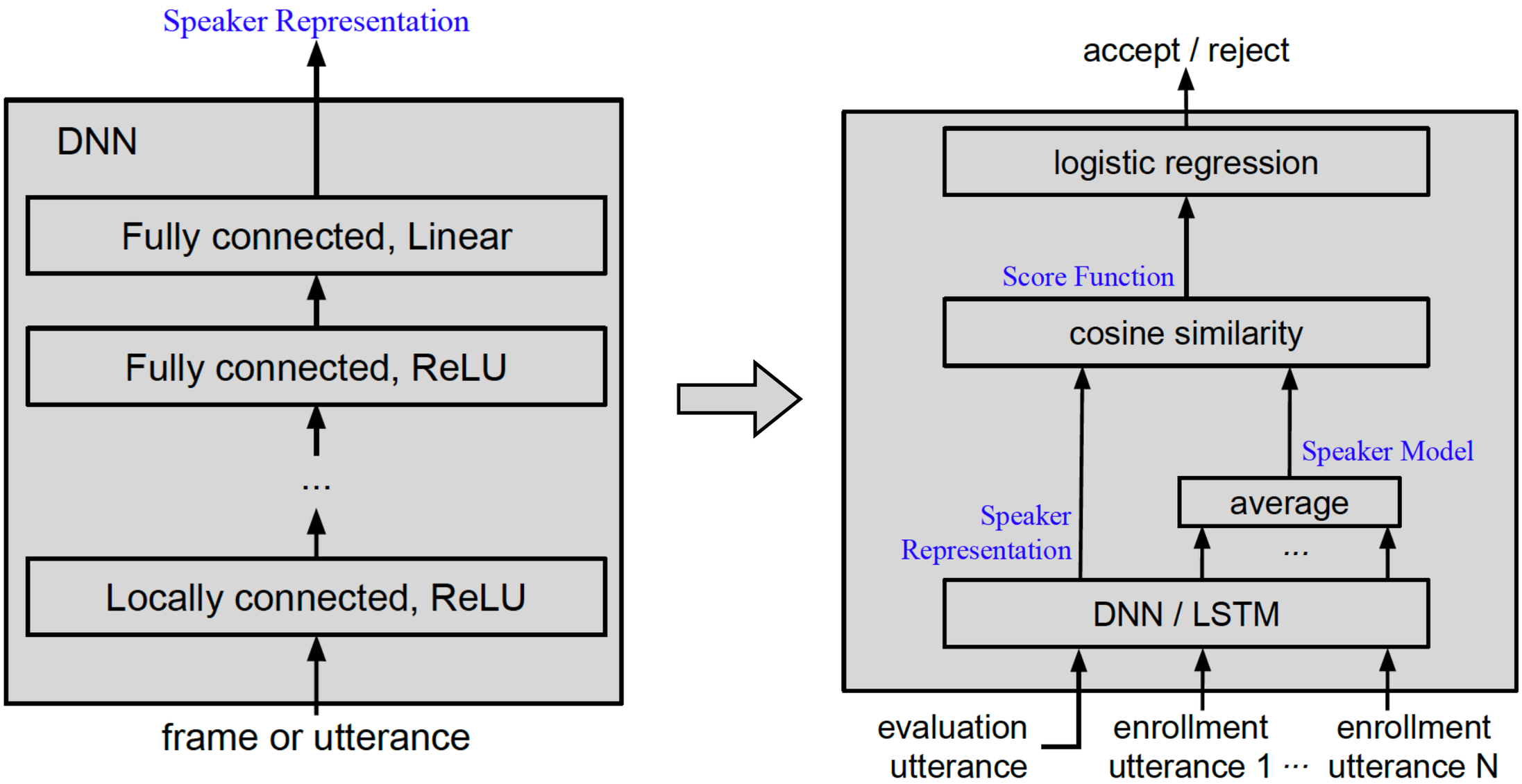


Deeper Model: Deep Speaker

- Model Training

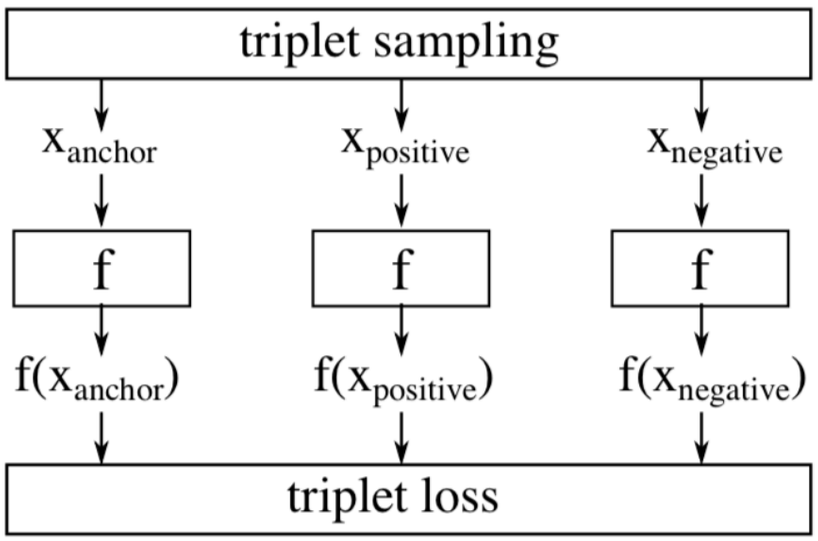


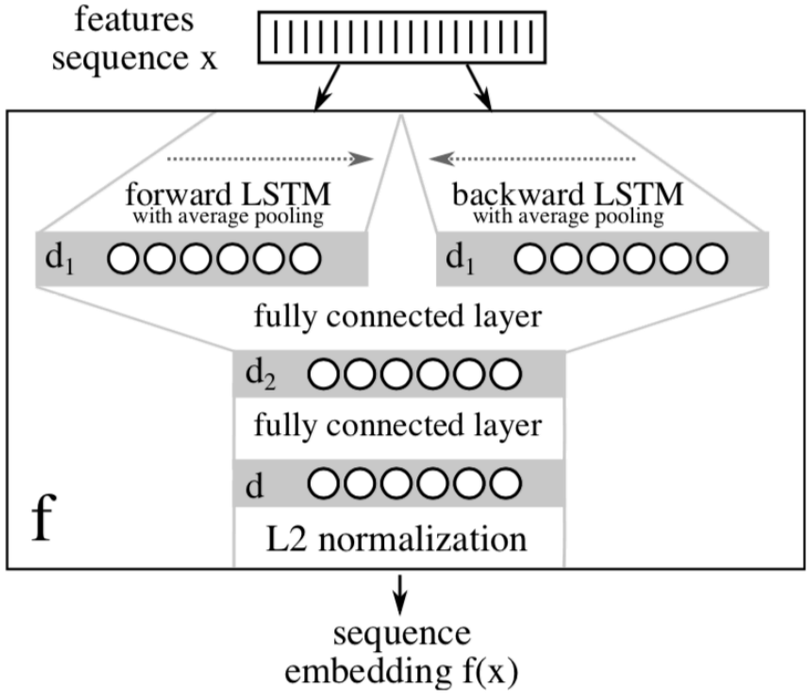
End-to-end SV


Speaker Representation
DNN
Fully connected, Linear
Fully connected, ReLU
...
Locally connected, ReLU
frame or utterance
accept / reject
logistic regression
Score Function
cosine similarity
Speaker Model
Speaker Representation
average
...
DNN / LSTM
evaluation utterance
enrollment utterance 1 ···
enrollment utterance N

Deep Model: Deep Metric Learning

Map speech sequences into sequence embedding space which utilizes Euclidean distance to measure similarity.





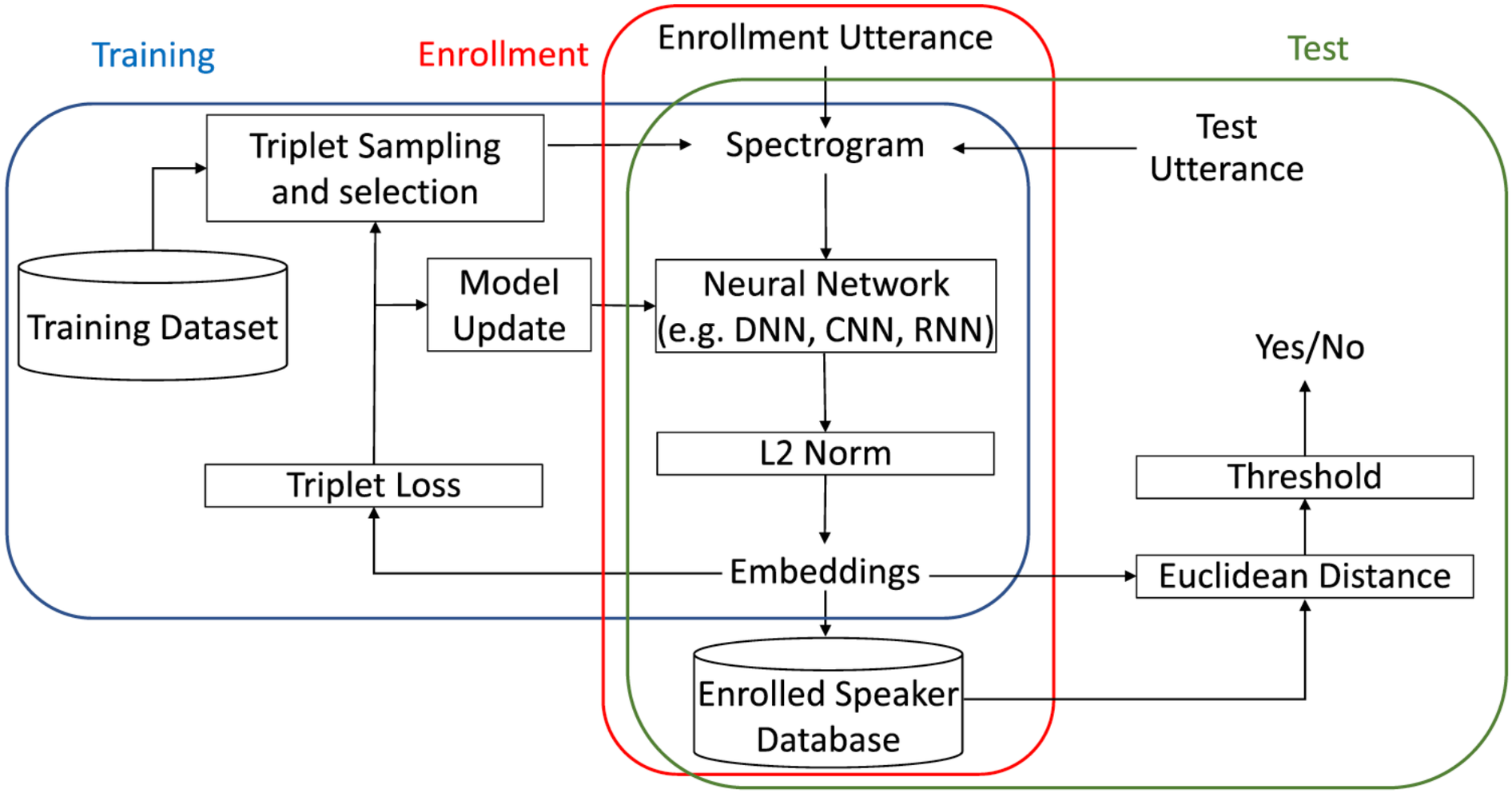
Deep Metric learning: Triplet Loss

- Loss function:

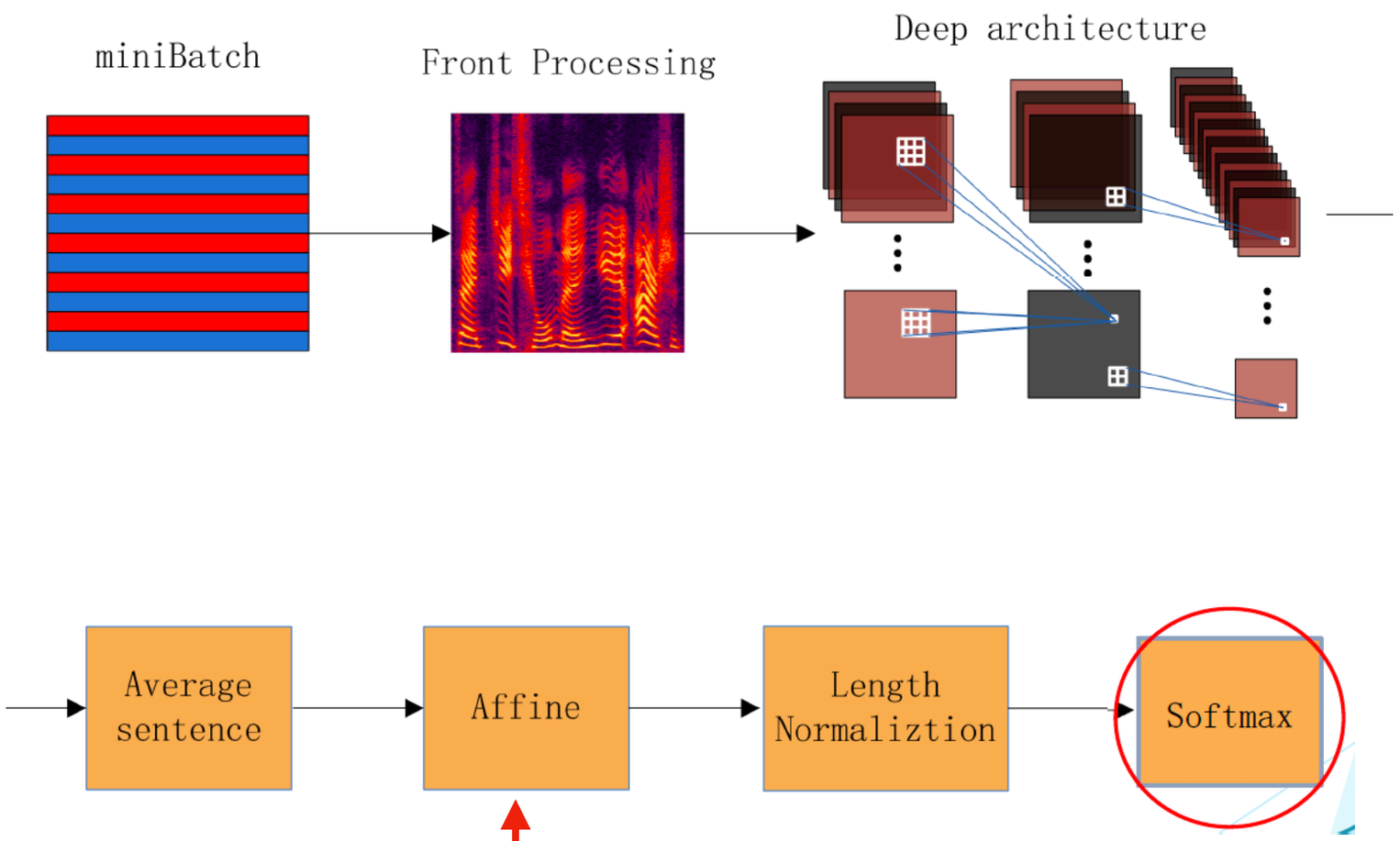
$$\Delta_\tau = \|f(x_a^\tau) - f(x_p^\tau)\|_2^2 - \|(f(x_a^\tau) - f(x_n^\tau)\|_2^2$$

$$\mathcal{L}(\mathcal{T}) = \sum_{\tau \in \mathcal{T}} \max(0, \Delta_\tau + \alpha)$$

DML SV system



Deep Model: Deep Speaker Embeddings



Summary on Deep Models

- Pros

  Easy and simple to reach comparable results as traditional method.

- Cons

  1. Eager for data.

  2. Lack of explanation. (Common problem)