简洁的说话人识别及语音识别

Toward Simple Speaker Recognition and Speech Recognition

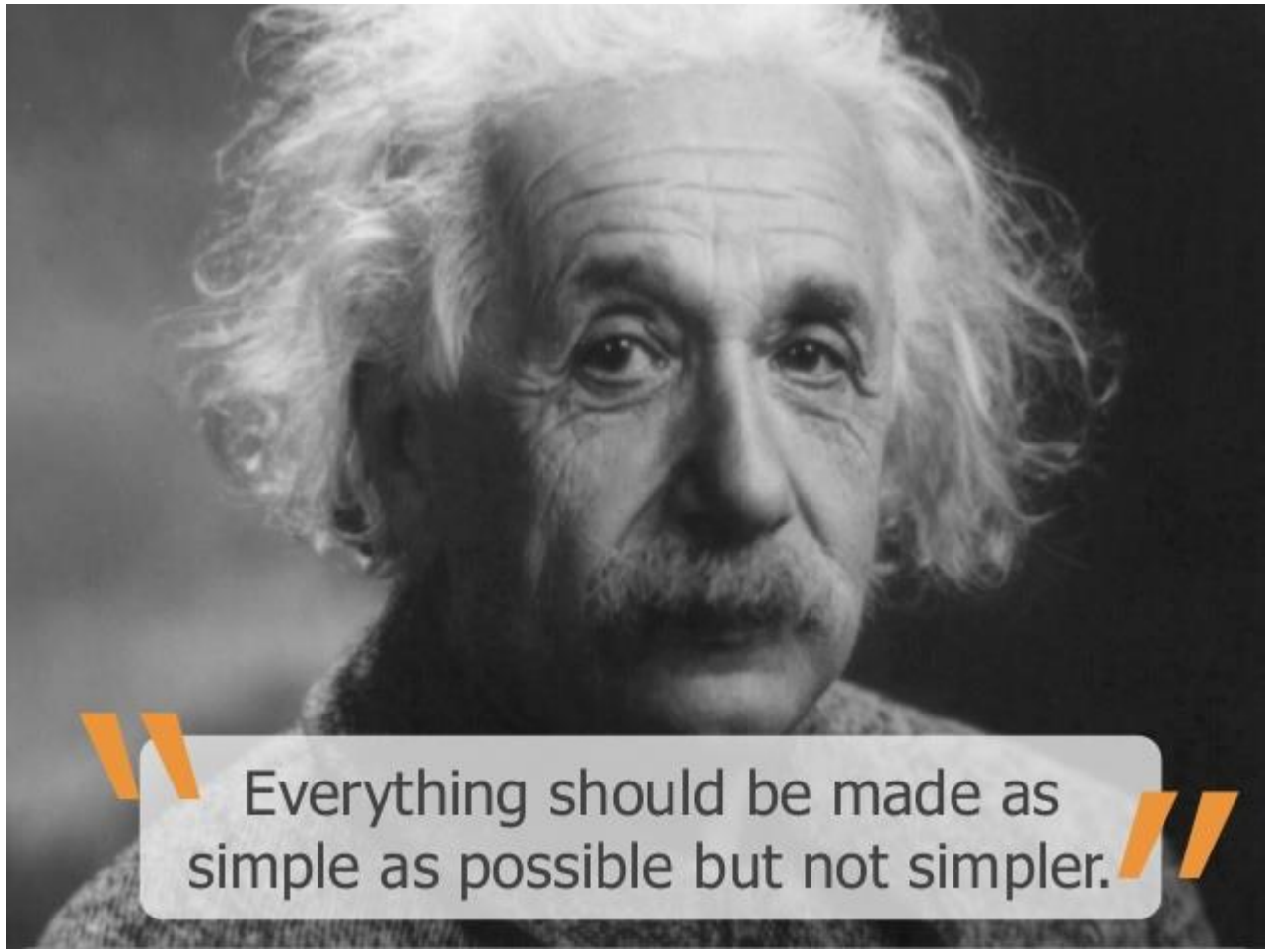
欧智坚

Speech Processing and Machine Intelligence (SPMI) Lab

Tsinghua University

http://oa.ee.tsinghua.edu.cn/ouzhijian/

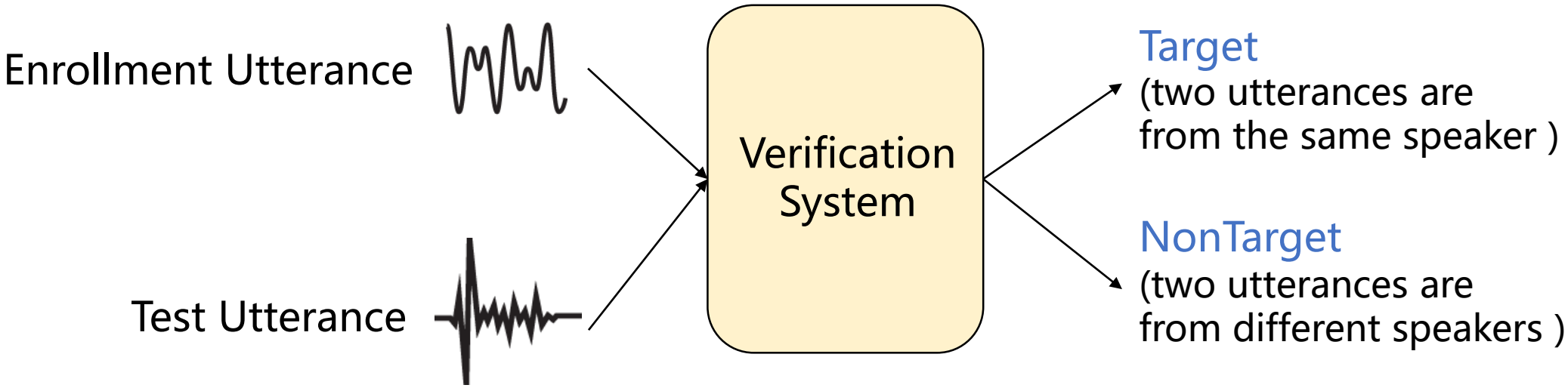
Motivation


Everything should be made as simple as possible but not simpler.

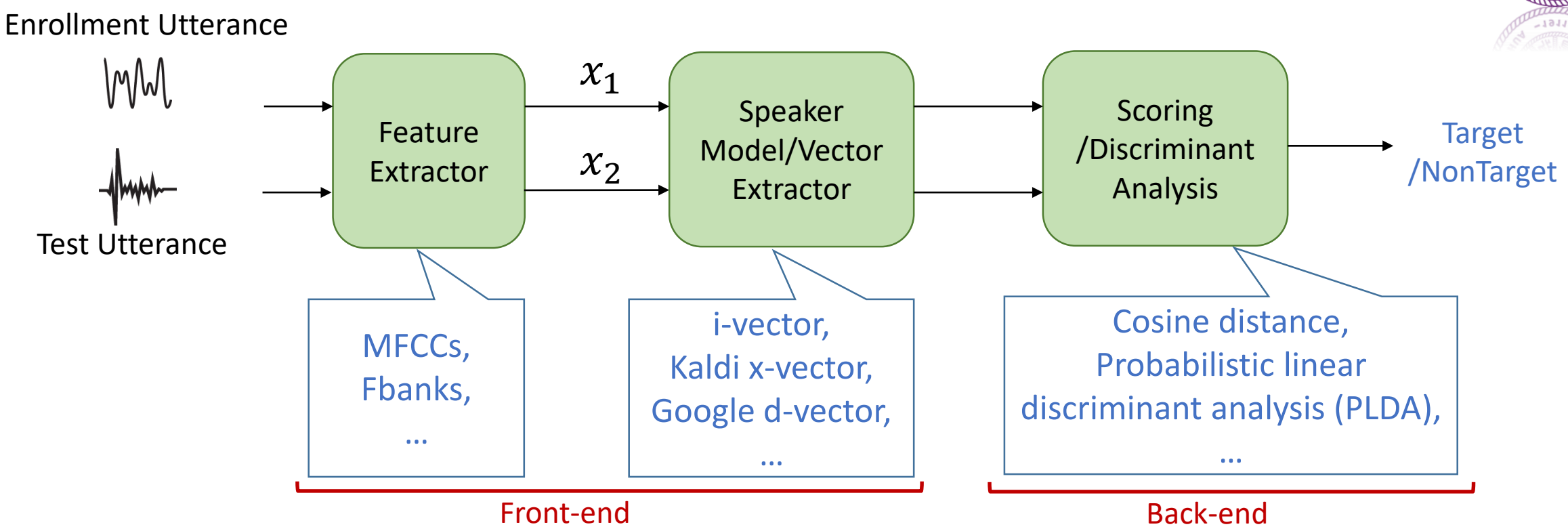
Content

- Angular Softmax Loss for End-to-end Speaker Verification
  - ISCSLP2018
- Joint Bayesian Gaussian discriminant analysis for speaker verification
  - ICASSP2017
- CRF-based Single-stage Acoustic Modeling with CTC Topology
  - ICASSP2019

What is text-independent speaker verification?



Speaker verification pipeline



Optimal binary decision by Likelihood Ratio Test (LRT):

$$Score(x_1, x_2) = log \frac{P(x_1, x_2|Target)}{P(x_1, x_2|NonTarget)}$$

1. The GMM-UBM method

— a generative approach

$$\frac{P(x_1, x_2|Target)}{P(x_1, x_2|NonTarget)} = \frac{P(x_2|Target,\ x_1)}{P(x_2|NonTarget, x_1)} \frac{P(x_1|Target)}{P(x_1|NonTarget)}$$

$$= \frac{P(x_2|GMM(x_1))}{P(x_2|NonTarget)}$$

MAP adaptation from UBM using x_1

$$= \frac{P(x_2|GMM(x_1))}{P(x_2|UBM)}$$

- GMM: Gaussian Mixture Model
- UBM: Universal Background Model
- MAP: Maximum A-Posteriori

Reynolds, "An Overview of Automatic Speaker Recognition Technology", ICASSP 2002.

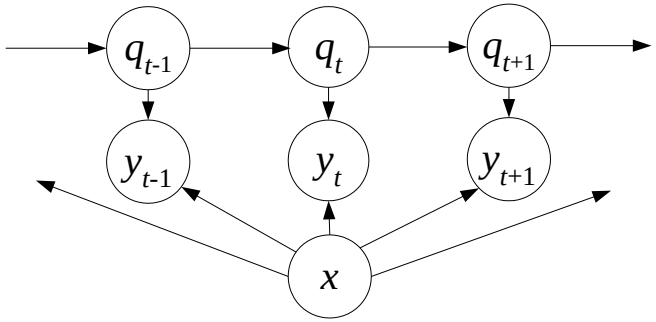
2. The i-vector method

— a generative speaker-vector extraction + a discriminative scoring

+ a generative scoring (e.g. PLDA)

$$log \frac{P(x_1, x_2|Target)}{P(x_1, x_2|NonTarget)} = cos[ivector(x_1), ivector(x_2)] + const$$

Train a generative model
(basically a factor-analysis model)
to extract speaker embeddings



Gaussian mean supervector

$\mu = \mu_0 + \Gamma x$

- Interestingly, generative scoring better than discriminative scoring

Dehak, et al., “Front-end factor analysis for speaker verification”, T-ASLP, 2011.

Ou and Luo, "Latent correlation analysis of HMM parameters for speech recognition", ICASSP 2007.

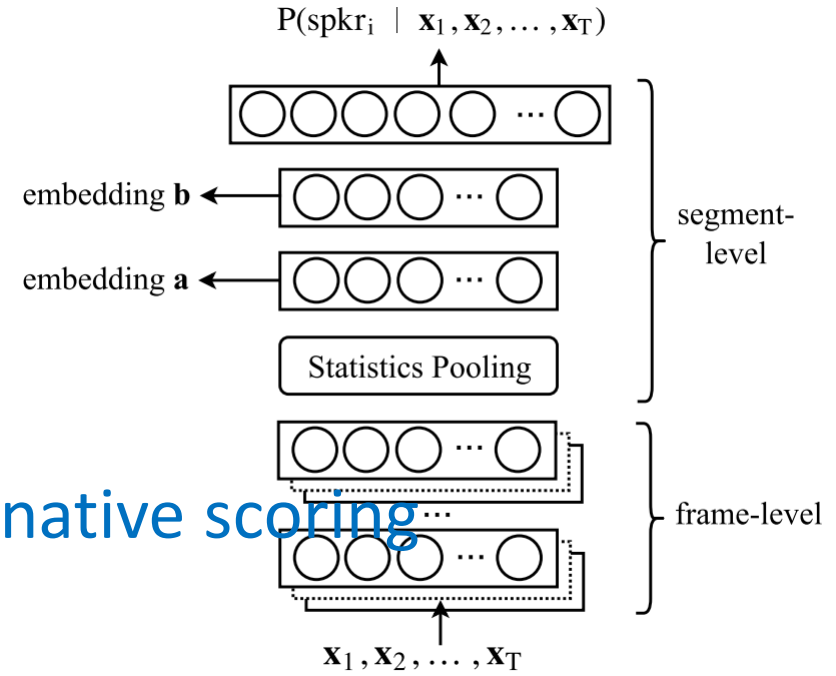
3. The x-vector method

— a discriminative speaker-vector extraction + a discriminative scoring

+ a generative scoring (e.g. PLDA)

$$log\frac{P(x_1,x_2|Target)}{P(x_1,x_2|NonTarget)} = cos[xvector(x_1), xvector(x_2)] + const$$

Train a discriminative model (often a neural network) to extract speaker embeddings



- Interestingly, generative scoring better than discriminative scoring

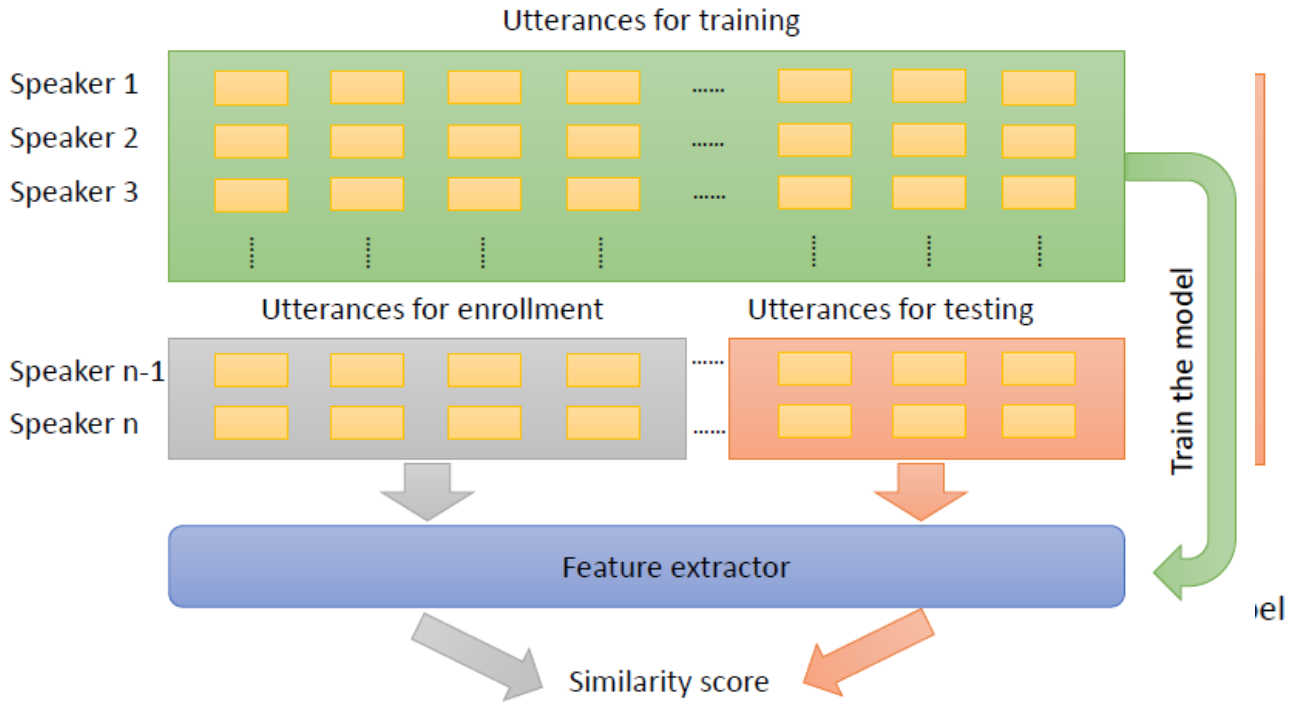
D. Snyder, et al., "Deep neural network embeddings for text-independent speaker verification", Interspeech 2017.

Angular softmax training

Yutian Li, Feng Gao, Zhijian Ou, Jiasong Sun.
Angular Softmax Loss for End-to-end Speaker Verification.
ISCSLP 2018. [Best Student Paper Award]

Motivation

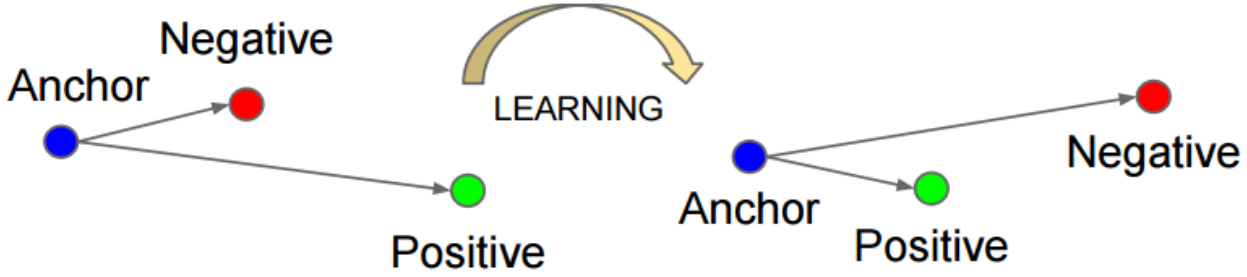
- Explore different criteria (loss functions)
- The softmax loss is more suitable for classification tasks
- Verification is different from classification



Losses for end-to-end model

- Triplet Loss: Make the Anchor-Positive pair closer and the Anchor-Negative pair farther.

$$L_{Triplet} = [\|f(x^a) - f(x^p)\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + \alpha]_+$$



- Requires a careful triplet selection procedure
  - ➢ both time-consuming and performance-sensitive.
- Training with triplet loss remains to be a difficult task
  - ➢ Efforts: generating triplets online from within a mini-batch (Schroff, et al), doing softmax pre-training (Li, et al).

Schroff, et al. “Facenet: A unified embedding for face recognition and clustering,” in CVPR, 2015.

Li, et al. "Deep speaker: an end-to-end neural speaker embedding system." *arXiv preprint* (2017).

Brief introduction for Angular Softmax Loss

- Recently proposed to improve softmax loss in face verification problem (Liu Weiyang, et al.).
- Introduces a margin between the target class and the non-target class into the softmax loss
- Drive end-to-end training of neural networks to learn angularly discriminative features
- Outperforms Softmax Loss, Triplet Loss ,Center Loss and Contrastive Loss.

Liu, Weiyang, et al. "Sphereface: Deep hypersphere embedding for face recognition." *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

A motivating binary classification example

- Softmax calculates the posterior probability for two classes as

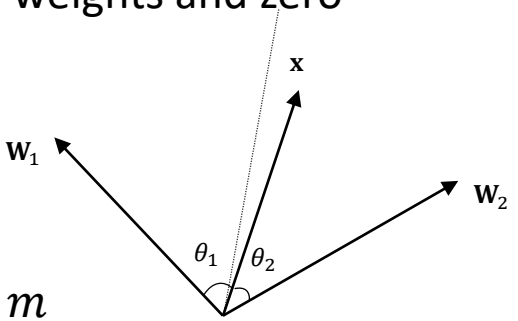
$$p_1 = \frac{\exp(\mathbf{W}_1^T\mathbf{x} + b_1)}{\exp(\mathbf{W}_1^T\mathbf{x} + b_1) + \exp(\mathbf{W}_2^T\mathbf{x} + b_2)} \qquad p_2 = \frac{\exp(\mathbf{W}_2^T\mathbf{x} + b_2)}{\exp(\mathbf{W}_1^T\mathbf{x} + b_1) + \exp(\mathbf{W}_2^T\mathbf{x} + b_2)}$$

- The decision boundary produced by Softmax Loss is

$$(\mathbf{W}_1 - \mathbf{W}_2)\mathbf{x} + b_1 - b_2 = 0$$

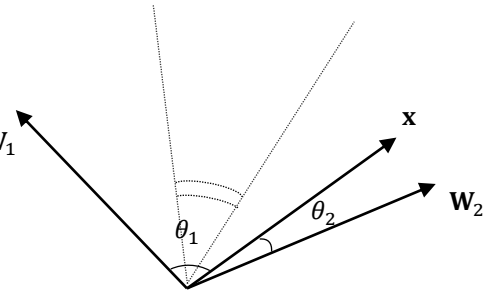
- To achieve angular decision boundary, we normalize the weights and zero out the biases. The decision boundary becomes

$$\|\mathbf{x}\|(\cos(\theta_1) - \cos(\theta_2)) = 0$$



- Further introduce angular margin, an integer parameter m

$$\Rightarrow \|\mathbf{x}\|(\cos(\theta_1) - \cos(\theta_2)) > 0$$
$$\|\mathbf{x}\|(\cos(m\theta_1) - \cos(\theta_2)) > 0 \quad \textit{for class 1}$$
$$\|\mathbf{x}\|(\cos(\theta_1) - \cos(m\theta_2)) < 0 \quad \textit{for class 2}$$
$$\Rightarrow \|\mathbf{x}\|(\cos(\theta_1) - \cos(\theta_2)) < 0$$



- Larger m means stronger discriminative ability

Definition of Angular Softmax Loss

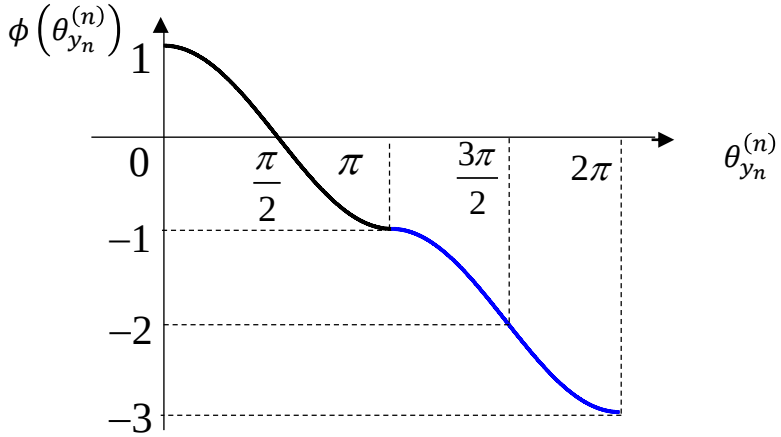
- Angular Softmax Loss, defined over training samples $\left(\mathbf{x}^{(n)}, y^{(n)}\right), n = 1, \cdots, N$

$$L_{ang} = \frac{1}{N}\sum_{n=1}^{N} -\log\left(\frac{e^{\|\mathbf{x}^{(n)}\| cos\left(m\theta_{y_n}^{(n)}\right)}}{e^{\|\mathbf{x}^{(n)}\| cos\left(m\theta_{y_n}^{(n)}\right)} + \sum_{j\neq y_n} e^{\|\mathbf{x}^{(n)}\| \cos\left(\theta_j^{(n)}\right)}}\right)$$

$$L_{ang} = \frac{1}{N}\sum_{n=1}^{N} -\log\left(\frac{e^{\|\mathbf{x}^{(n)}\| \phi\left(\theta_{y_n}^{(n)}\right)}}{e^{\|\mathbf{x}^{(n)}\| \phi\left(\theta_{y_n}^{(n)}\right)} + \sum_{j\neq y_n} e^{\|\mathbf{x}^{(n)}\| \cos\left(\theta_j^{(n)}\right)}}\right)$$

where $\phi\left(\theta_{y_n}^{(n)}\right) = (-1)^k \cos\left(m\theta_{y_n}^{(n)}\right) - 2k$ for $\theta_{y_n}^{(n)} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right]$, $k \in [0, m-1]$

- Examples of $\phi\left(\theta_{y_n}^{(n)}\right)$ with m=2



Experimental Settings

- Data
  - Training, subset of Fisher data, 5000 speakers
  - Evaluation, subset of Fisher data, 1000 speakers
  - No overlap between training and test data

Table: Number of speakers

	Total	Male	Female
Training	5000	2500	2500
Evaluation	1000	500	500

- Network architecture
  - For comparison, the network architecture is similar to Kaldi x-vector.

utterance level layers	FC_2 (512→300)
	FC_1 (3000→512)
statistic pooling layer	mean and standard deviation
frame level layers	TDNN_5 (512×1→1500)
	TDNN_4 (512×1→512)
	TDNN_3 (512×3→512)
	TDNN_2 (512×3→512)
	TDNN_1 (23×5→512)

Experimental Results

- Enrollment, 3000 frames.
- Test, varies in 300, 500, 1000, 1500 frames.
  - Using triplet loss yields inferior performance
  - For short test condition, Angular Softmax is better than baselines
  - Angular softmax outperforms softmax under the same network
  - Angular softmax with m=4 is not always the best

EER (%)		Durations of test utterances			
Model	Loss + metric	300	500	1000	1500
i-vector	-- + PLDA	1.00	0.53	0.33	0.37
x-vector	Softmax + PLDA	1.86	0.83	0.40	0.43
Our Network	Softmax + PLDA	1.30	0.97	0.70	0.73
	Angular m=2 + Cosine	0.94	0.60	0.47	0.57
	Angular m=3 + Cosine	0.67	0.40	0.37	0.43
	Angular m=4 + Cosine	0.70	0.47	0.33	0.47
	Triplet + Cosine	2.17	1.63	1.17	1.23

Experimental Results

- Enrollment and test, both vary in 300, 500, 1000, 1500 frames.
  - For short utterance condition, using PLDA back-end significantly reduce EERs of the Angular softmax systems.
  - For short test condition, Angular Softmax is better than baselines
  - Angular softmax outperforms softmax under the same network

EER (%)		Durations of utterances			
Model	Loss + metric	300	500	1000	1500
i-vector	-- + PLDA	2.93	1.57	0.50	0.47
x-vector	Softmax + PLDA	3.17	1.63	0.63	0.63
Our Network	Softmax + PLDA	3.43	2.40	1.20	1.07
	Angular m=2 + Cosine	2.90	1.57	0.77	0.83
	Angular m=2 + PLDA	2.17	1.33	0.73	0.80
	Angular m=3 + Cosine	2.50	1.23	0.73	0.56
	Angular m=3 + PLDA	2.10	1.33	0.70	0.77
	Angular m=4 + Cosine	2.43	1.33	0.70	0.63
	Angular m=4 + PLDA	2.23	1.37	0.73	0.90

Further results on SRE-18

- We use Angular Softmax Loss in SRE-18, CMN2 (Call My Net 2) test dataset.

- Training set for neural network consists of :
  - Fisher
  - Switchborad
  - Previous SRE evaluation data
  - Mixer 6
  - Voxceleb

- The result in SRE-18 CMN2 development data.

Model + metric	EER (%)
Kaldi i-vector + PLDA	15.08
Kaldi Softmax + PLDA	9.64
Angular m=2 + PLDA	9.46
Angular m=3 + PLDA	9.93

Contributions and Conclusions

Two contributions:

- We introduce Angular Softmax Loss into end-to-end speaker verification
  - ➢ Can learn more discriminative features than softmax and triplet loss;
  - ➢ Easy and stable for usage.

- The combination of using Angular softmax in training the front-end and using PLDA in the back-end scoring further boosts the performance.

Conclusions:

- Angular Softmax performs better than Kaldi i-vector baseline for short test utterances.

- In SRE18 development dataset, Angular Softmax x-vector outperforms both Kaldi i-vector and x-vector baselines.

Joint Bayesian Discriminant Analysis

Yiyan Wang, Haotian Xu, Zhijian Ou.
Joint Bayesian Gaussian discriminant analysis for speaker verification.
ICASSP 2017.

Definition & Training

- The j-th speaker-vector of speaker i, denoted by $x_{ij} \in R^d, j = 1, \cdots, m_i$

is decomposed as ~~$x_{ij} = Fz_i + \varepsilon_{ij}$~~

$$x_{ij} = \mu_i + \varepsilon_{ij}$$

Within-speaker variable

Speaker identity variable

- Two independent Gaussians, $\mu_i \sim N(0, S_\mu)$ $\varepsilon_{ij} \sim N(0, S_\varepsilon)$
- The model parameters $\Theta = \{S_\mu, S_\varepsilon\}$ are estimated by **EM algorithm**

$$\max_{\Theta} \sum_i E_{p(h_i|x_i;\Theta^t)}[\log p(h_i; \Theta^{t+1})]$$

where $x_i = [x_{i1}; \cdots; x_{im_i}]$, $h_i = [\mu_i; \varepsilon_{i1}; \cdots; \varepsilon_{im_i}]$

Chen, et al., "An efficient joint formulation for Bayesian face verification", T-PAMI, 2016.

Log-likelihood ratio score & Efficient testing

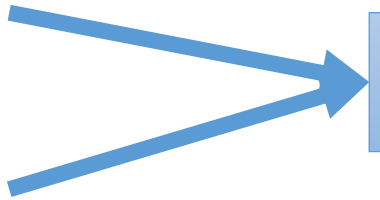
- The **log-likelihood ratio (LLR) score** between i-vectors x_1 and x_2 is

$$r(x_1, x_2) = log\frac{p(x_1, x_2|H_s)}{p(x_1, x_2|H_d)} = logp(x_1, x_2) - logp(x_1) - logp(x_2)$$

- Testing: do Simultaneous Diagonalization (SD) of S_μ and S_ε

$$\Phi^T S_\mu \Phi = K$$

$$\Phi^T S_\varepsilon \Phi = I$$



Diagonal matrix

Log-likelihood ratio score & Efficient testing

- Keep the first $s < d$ largest eigenvalues of $S_\mu^{-1} S_\varepsilon$, giving the low-rank diagonal matrix K.

- Define $\Psi = \Phi^{-T}$ $\longrightarrow$ $S_\mu = \Psi K \Psi^T$ $\quad$ $S_\varepsilon = \Psi I \Psi^T$

$$\Sigma_{x_i} = \begin{bmatrix} S_\mu + S_\varepsilon & S_\mu & \cdots & S_\mu \\ S_\mu & S_\mu + S_\varepsilon & \cdots & S_\mu \\ \vdots & \vdots & \ddots & \vdots \\ S_\mu & S_\mu & S_\mu & S_\mu + S_\varepsilon \end{bmatrix} = \Omega \begin{bmatrix} K + I & K & \cdots & K \\ K & K + I & \cdots & K \\ \vdots & \vdots & \ddots & \vdots \\ K & K & K & K + I \end{bmatrix} \Omega^T$$

where $\Omega = diag(\Psi; \cdots; \Psi)$

- The calculation of $p(x_i)$ could be accelerated, only involves inversion of diagonal matrices

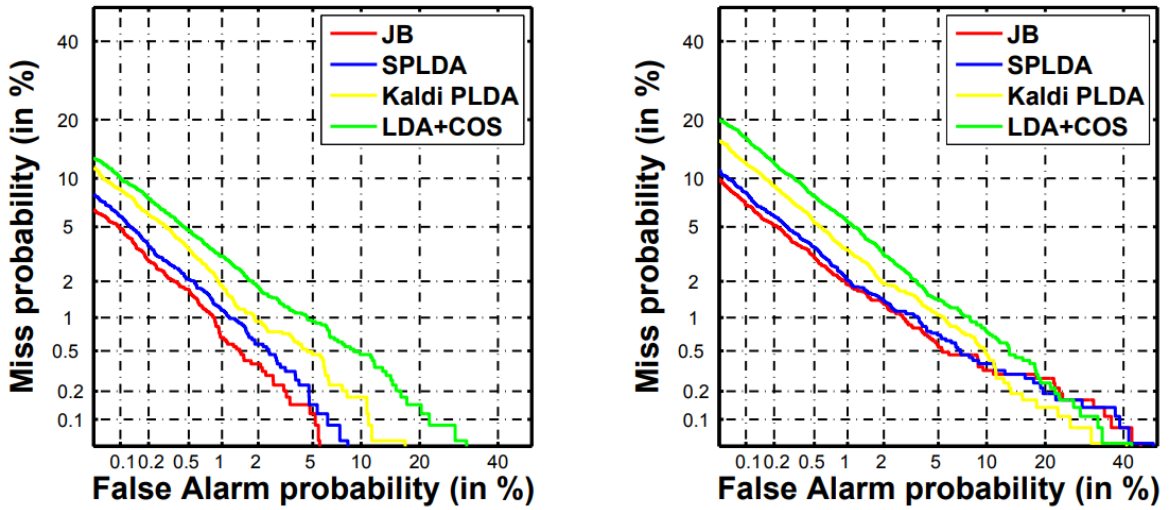
Complexity: $O(d^3) \rightarrow O(d)$

Experimental Results

- Speaker verification performance
  - Experiment on NIST SRE10 dataset, using i-vectors
  - Compared with LDA+Cosine distance, SPLDA, Kaldi PLDA
  - Results: beat all other scoring methods

System	SRE10 MALE			SRE10 FEMALE		
	EER	DCF10	DCF08	EER	DCF10	DCF08
LDA+COS	1.905	0.292	0.091	2.619	0.399	0.126
Kaldi PLDA	1.299	0.284	0.079	1.944	0.345	0.102
SPLDA	1.010	0.217	0.055	1.621	0.287	0.079
JB	**0.894**	**0.188**	**0.048**	**1.485**	**0.245**	**0.069**

Table 3. Performance comparison of four different discriminant analysis back-ends on NIST SRE10 core condition 5.



(a) SRE10 MALE (b) SRE10 FEMALE

Fig. 1. DET curves for JB, SPLDA, Kaldi PLDA and LDA in SRE10 core condition 5 evaluation.

Experimental Results

- Reduce subspace dimensionality in testing
  - SPLDA is sensitive with the subspace dimensionality in testing
  - The JB performance fluctuates slightly, more robust
  - SVD (Chen et al) and SD have close performances but SD has wider applicability

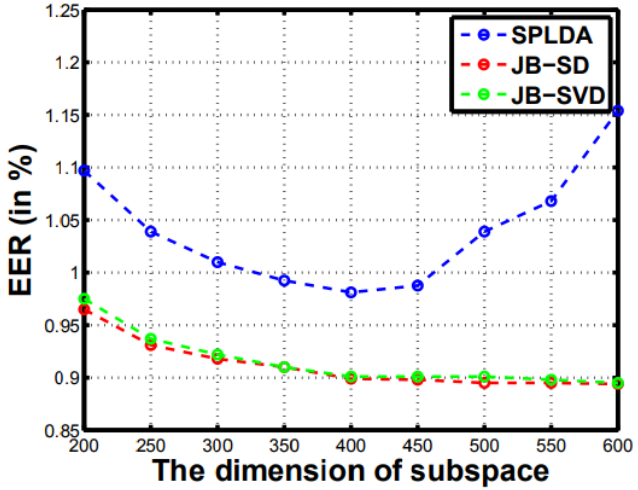


Fig. 2. The influence of subspace dimensionality on JB and SPLDA using NIST SRE10 core condition male test data.

Chen, et al., "An efficient joint formulation for Bayesian face verification", T-PAMI, 2016.

Contributions and Conclusions

We apply JB to speaker verification and make three contributions beyond the original JB.

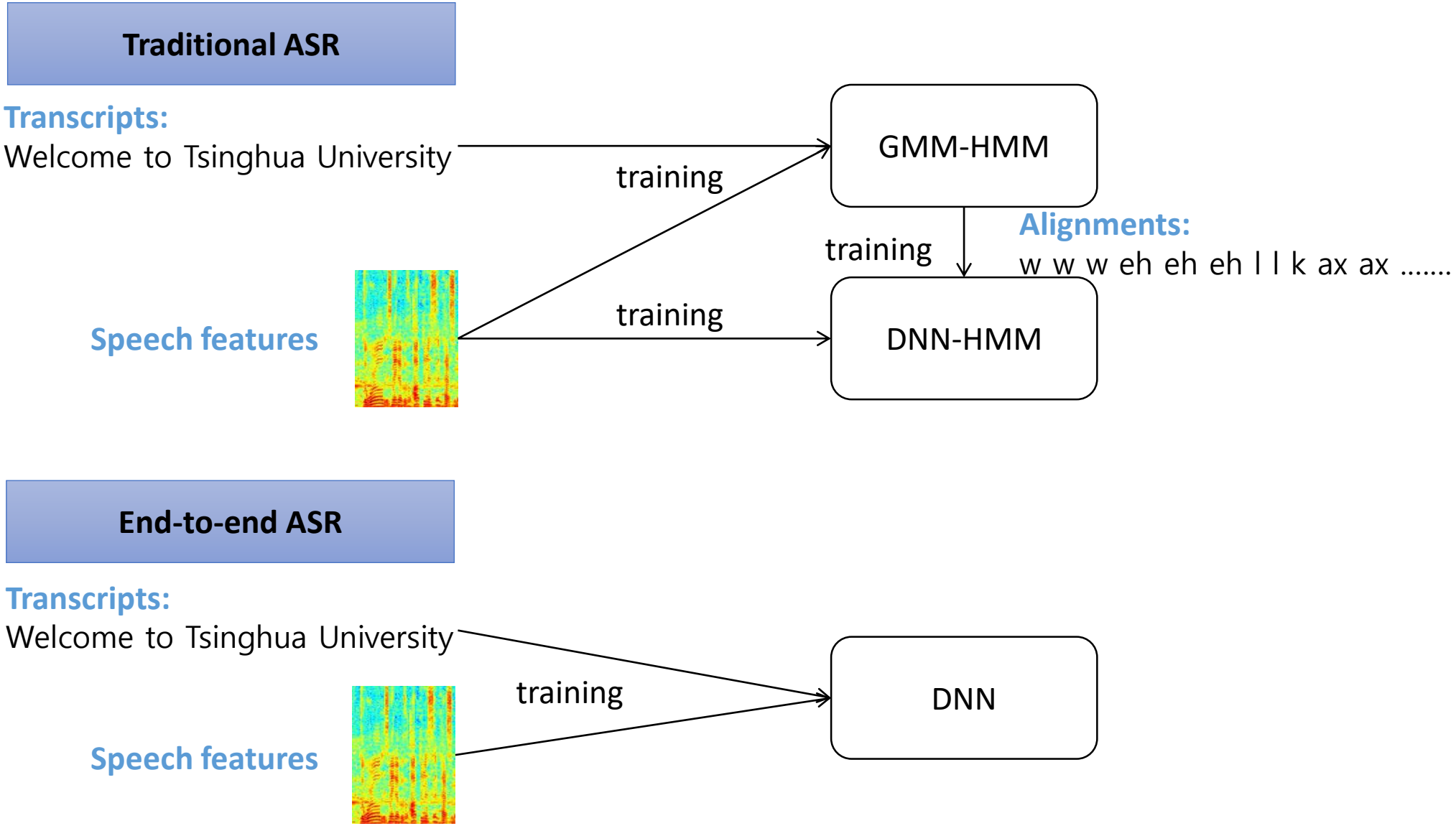
1. EM iterations with exact statistics.
2. Simultaneous diagonalization (SD) to achieve efficient testing.
3. Theoretically scrutinize similarities and differences between various Gaussian-based discriminant analysis.

Method	JB	two-covariance	SPLDA	Kaldi PLDA
Observation	$x_i = \{x_{ij}, j = 1, \ldots, m_i\}$			$\bar{x}_i = \frac{1}{m_i}\sum_{j=1}^{m_i} x_{ij}$
Model	$x_{ij} = \mu_i + \varepsilon_{ij}$		$x_{ij} = Fz_i + \varepsilon_{ij}$	$\bar{x}_i = \mu_i + \varepsilon_{i1}$
h_i	$\{\mu_i, \{\varepsilon_{ij}\}\}$	$\{\mu_i\}$	$\{z_i\}$	$\{\mu_i, \varepsilon_{i1}\}$
EM objective function $Q(\Theta_t, \Theta_{t+1})$	$E_{p(h_i\|x_i)}[logp(h_i)]$	$E_{p(h_i\|x_i)}[logp(x_i, h_i)]$		$E_{p(h_i\|\bar{x}_i)}[logp(h_i)]$
Subspace dimensionality setting	loose		strict	loose
EM convergence	fast	slow		fast

CTC-CRF

Hongyu Xiang, Zhijian Ou.
“CRF-based single-stage acoustic modeling with CTC topology”,
ICASSP 2019. [Oral]

Introduction

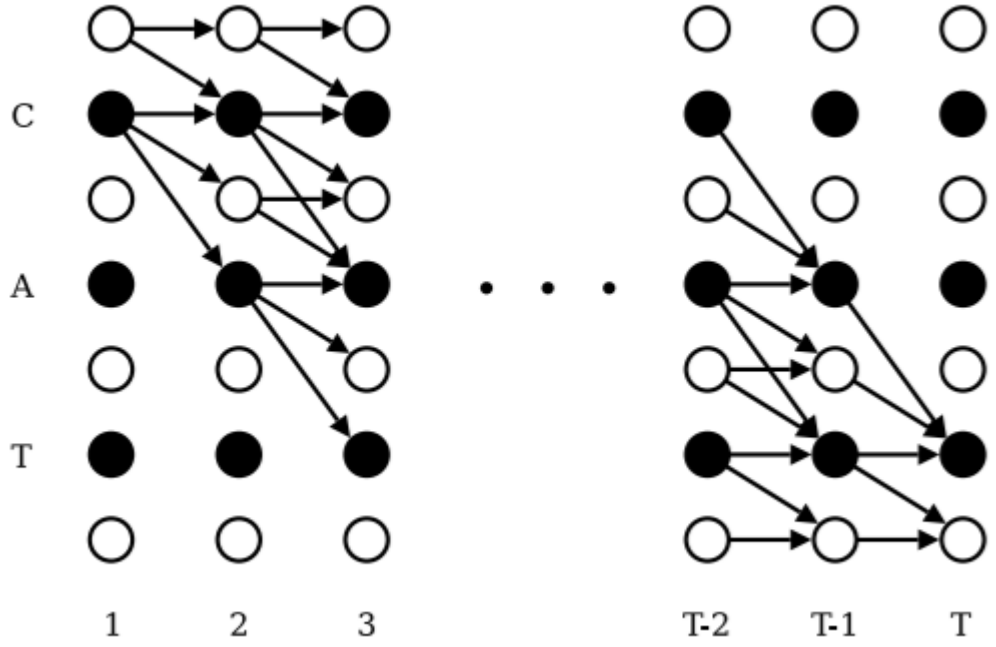


Introduction

- End-to-end system:
  - Eliminate GMM-HMM pre-training and tree building, and can be trained from scratch (flat-start or single-stage).
- In a more strict sense:
  - Remove the need for a pronunciation lexicon and, even further, train the acoustic and language models jointly rather than separately. Data-hungry.

We are interested in advancing single-stage acoustic models, which use a separate language model (LM) with or without a pronunciation lexicon. Data-efficient.

CTC



Alex Graves. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. ICML, 2006.

Graves CTC = HMM-DNN with a special state topology

Speech feature: $\boldsymbol{x}$
Label sequence: $\boldsymbol{l}$
State sequence: $\boldsymbol{\pi}$

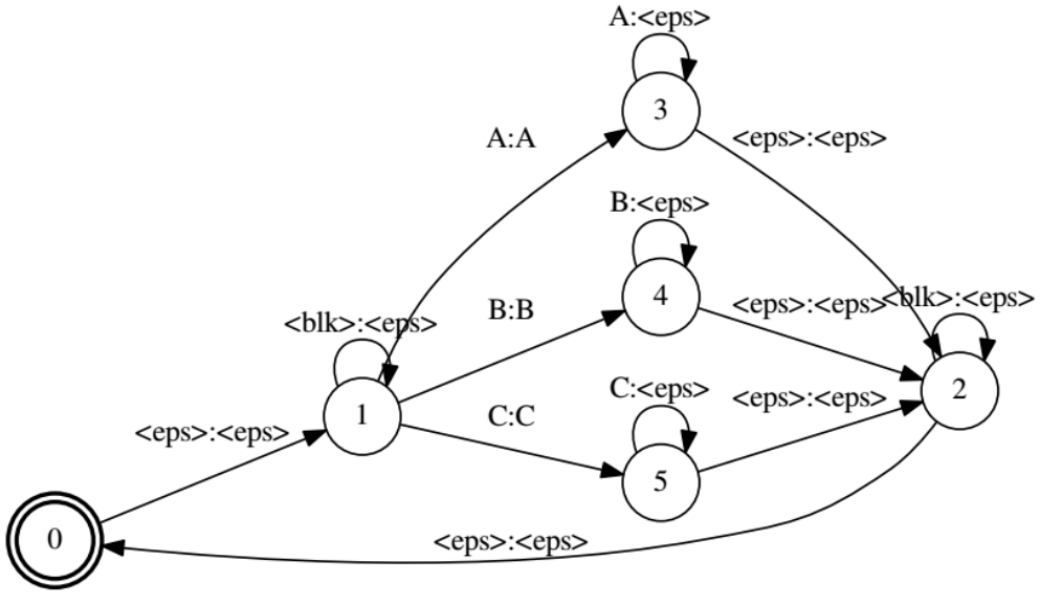
A mapping $\mathcal{B}$ maps $\boldsymbol{\pi}$ to $\boldsymbol{l}$ by:
1. removing all repetitive symbols between the blank symbols.
2. removing all blank symbols.

$$\mathcal{B}(-CC--AA-T-) = CAT$$

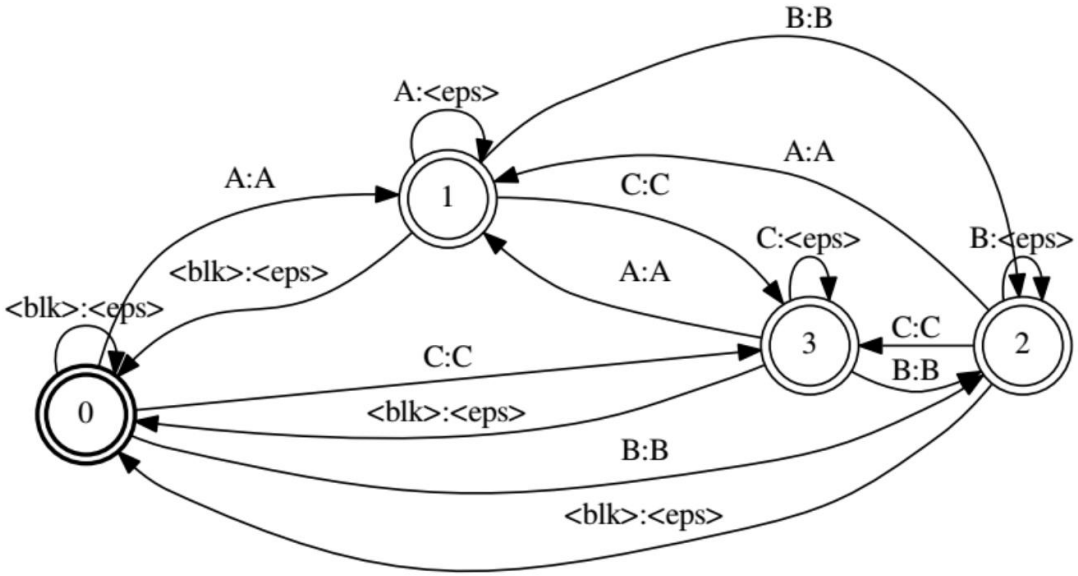
The probability of $\boldsymbol{l}$ given $\boldsymbol{x}$ is the sum of the probability of all corresponding $\boldsymbol{\pi}$ given $\boldsymbol{x}$

$$p(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{l})} p(\boldsymbol{\pi}|\boldsymbol{x})$$

WFST representation of CTC



Eesen T.fst

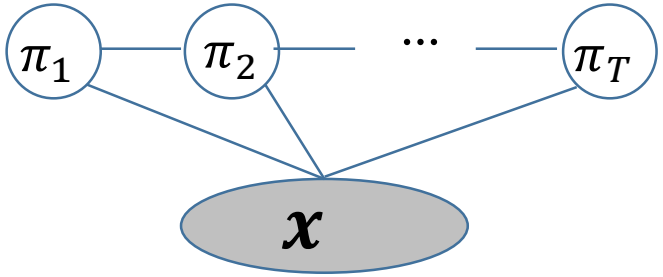


Corrected T.fst

CTC-CRF

- **CRF**: The conditional probability of the hidden state sequence $\boldsymbol{\pi}$ given the observation sequence $\boldsymbol{x}$ is defined as

$$p(\boldsymbol{\pi}|\boldsymbol{x}) = \frac{e^{\phi(\boldsymbol{\pi},\boldsymbol{x};\boldsymbol{\theta})}}{\sum_{\boldsymbol{\pi}'} e^{\phi(\boldsymbol{\pi}',\boldsymbol{x};\boldsymbol{\theta})}}$$



- **CTC**: If $\mathcal{B}$ maps $\boldsymbol{\pi}$ to a unique $\boldsymbol{l}$, then

$$p(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}} p(\boldsymbol{\pi}, \boldsymbol{l}|\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{l})} p(\boldsymbol{\pi}|\boldsymbol{x}; \boldsymbol{\theta})$$

We use the CTC mapping as $\mathcal{B}$.

CTC-CRF

- **Potential function**:

$$\phi(\boldsymbol{\pi}, \boldsymbol{x}; \boldsymbol{\theta}) = \sum_{t=1}^{T} \log p(\pi_t|\boldsymbol{x}) + \log p_{LM}\left(B(\boldsymbol{\pi})\right)$$

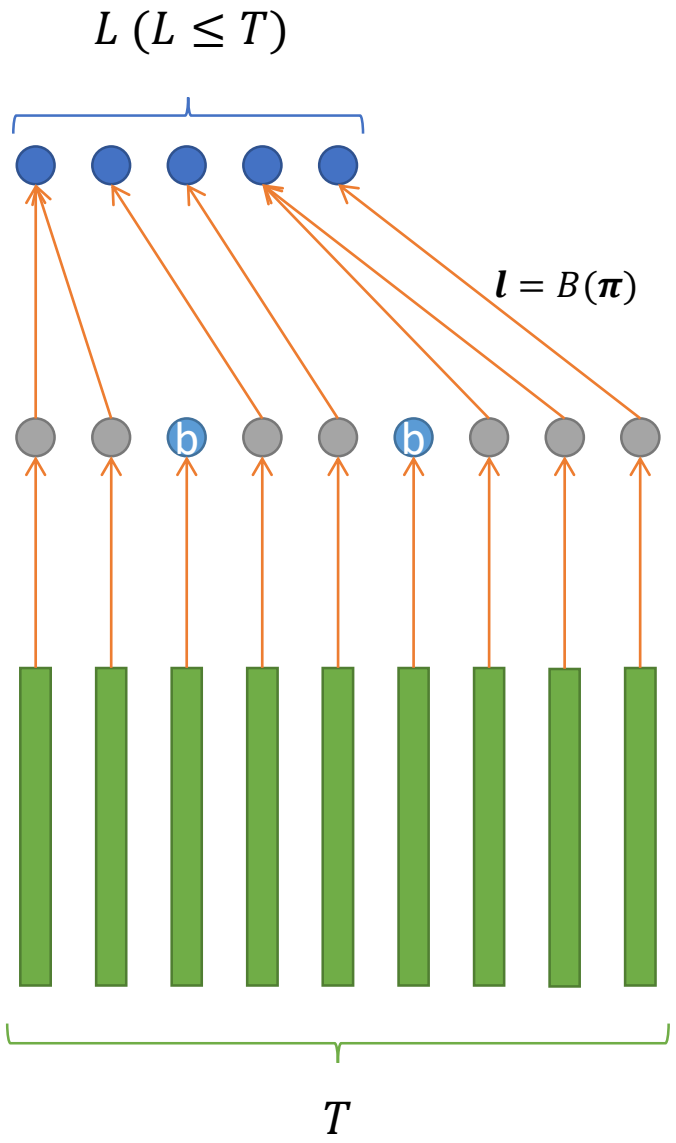
  - $\log p(\pi_t|\boldsymbol{x})$ is the node potential, calculated by the neural network.
  - $\log p_{LM}\left(B(\boldsymbol{\pi})\right)$ is the edge potential, calculated by n-gram denominator LM of labels, like in LF-MMI.

- **Objective function and Gradient**:

$$\mathcal{J}_{CRF}(\boldsymbol{\theta}) = \log p(\boldsymbol{l}|\boldsymbol{x}; \boldsymbol{\theta})$$

$$\frac{\partial \mathcal{J}_{CRF}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_{p(\boldsymbol{\pi}|\boldsymbol{l},\boldsymbol{x};\boldsymbol{\theta})}\left[\frac{\partial \phi(\boldsymbol{\pi}, \boldsymbol{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] - \mathbb{E}_{p(\boldsymbol{\pi}'|\boldsymbol{x};\boldsymbol{\theta})}\left[\frac{\partial \phi(\boldsymbol{\pi}', \boldsymbol{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]$$

CTC-CRF



Label sequence $\boldsymbol{l}$

State sequence $\boldsymbol{\pi}$

Speech feature $\boldsymbol{x}$

$$p(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in B^{-1}(\boldsymbol{l})} p(\boldsymbol{\pi}|\boldsymbol{x})$$

$$p(\boldsymbol{\pi}|\boldsymbol{x}) = \frac{e^{\phi(\boldsymbol{\pi},\boldsymbol{x};\boldsymbol{\theta})}}{\sum_{\boldsymbol{\pi}'} e^{\phi(\boldsymbol{\pi}',\boldsymbol{x};\boldsymbol{\theta})}}$$

$$\phi(\boldsymbol{\pi}, \boldsymbol{x}; \boldsymbol{\theta}) = \sum_{t=1}^{T} \log p(\pi_t|\boldsymbol{x}) + \log p_{LM}(B(\boldsymbol{\pi}))$$

node potential

edge potential

Differences between CTC-CRF and End-to-end LF-MMI (EE-LM-MMI / SS-LF-MMI)

- MMI training of HMM is equivalent to conditional maximum likelihood (CML) training of CRF. Our model is similar to SS-LF-MMI. The main differences are:

	EE-LM-MMI	**CTC-CRF**
State topologies	HMM with two states	CTC
Silence labels	Using silence labels. Randomly inserting silence labels when estimating denominator LM.	No silence labels. Use blank to absorb the silence labels. No need to insert silence labels to transcripts.
Alignments	Normal alignments	Alignments with peaks. Speed up decoding by skipping blanks

Experiments

- We conduct our experiments on three benchmark datasets:
  - WSJ 80 hours
  - Switchboard 300 hours
  - Librispeech 1000 hours
- Acoustic model: 6 layer BLSTM with 320 hidden dim, 13M parameters
- Adam optimizer with an initial learning rate of 0.001, decreased to 0.0001 when cv loss does not decrease
- Implemented with Pytorch.
- Objective function (use the CTC objective function to help convergences):

$$\mathcal{J}_{CTC-CRF} + \alpha \mathcal{J}_{CTC}$$

- Decoding score function (use word-based language models, WFST based decoding):

$$\log p(\boldsymbol{l}|\boldsymbol{x}) + \beta \log p_{LM}(\boldsymbol{l})$$

Experiments (Comparison with CTC, phone based)

SP: speed perturbation for 3-fold data augmentation.
"Y" means using SP. "N" means not using SP.

WSJ results

Model	Unit	LM	SP	dev93	eval92
CTC	Mono-phone	4-gram	N	10.81%	7.02%
CTC-CRF	Mono-phone	4-gram	N	6.24%	3.90%

Relative 44.4%

Switchboard results

Model	Unit	LM	SP	SW	CH
CTC	Mono-phone	4-gram	N	12.9%	23.6%
CTC-CRF	Mono-phone	4-gram	N	11.0%	21.0%

Relative 14.7%

Librispeech results

Model	Unit	LM	SP	Dev Clean	Dev Other	Test Clean	Test Other
CTC	Mono-phone	4-gram	N	4.64%	13.23%	5.06%	13.68%
CTC-CRF	Mono-phone	4-gram	N	3.87%	10.28%	4.09%	10.65%

Relative 19.1%

Experiments (Comparison with SS-LF-MMI, phone based)

WSJ

Model	Unit	LM	SP	dev93	eval92
SS-LF-MMI	Mono-phone	4-gram	Y	6.3%	3.1%
SS-LF-MMI	Bi-phone	4-gram	Y	6.0%	3.0%
CTC-CRF	Mono-phone	4-gram	Y	6.23%	3.79%

Switchboard

Model	Unit	LM	SP	SW	CH
SS-LF-MMI	Mono-phone	4-gram	Y	11.0%	20.7%
SS-LF-MMI	Bi-phone	4-gram	Y	9.8%	19.3%
CTC-CRF	Mono-phone	4-gram	Y	10.3%	19.7%
Seq2Seq	Subword	LSTM	N	11.8%	25.7%

Relative 6.4%

Relative 4.8%

Librispeech

Model	Unit	LM	SP	Dev Clean	Dev Other	Test Clean	Test Other
LF-MMI	Tri-phone	4-gram	Y	-	-	4.28%	-
CTC-CRF	Mono-phone	4-gram	N	3.87%	10.28%	4.09%	10.65%
Seq2Seq	Subword	4-gram	N	4.79%	13.1%	4.82%	15.30%

Relative 4.4%

Zeyer, Irie, Schlter, and Ney, “Improved training of end-to-end attention models for speech recognition”, Interspeech 2018.

Conclusions

- We propose a framework for single-stage acoustic modeling based on CRFs with CTC topology (CTC-CRF).
- CTC-CRFs achieve strong results on WSJ, Switchboard and Librispeech datasets.
  - CTC can be significantly improved by CTC-CRF.
  - CTC-CRF significantly outperforms Attention-based Seq2Seq.
  - CTC-CRF outperforms the SS-LF-MMI, across all the three benchmarking datasets and in both cases of mono-phones and mono-chars.
  - Avoids some ad-hoc operation in SS-LF-MMI.

Summary

1. Joint Bayesian

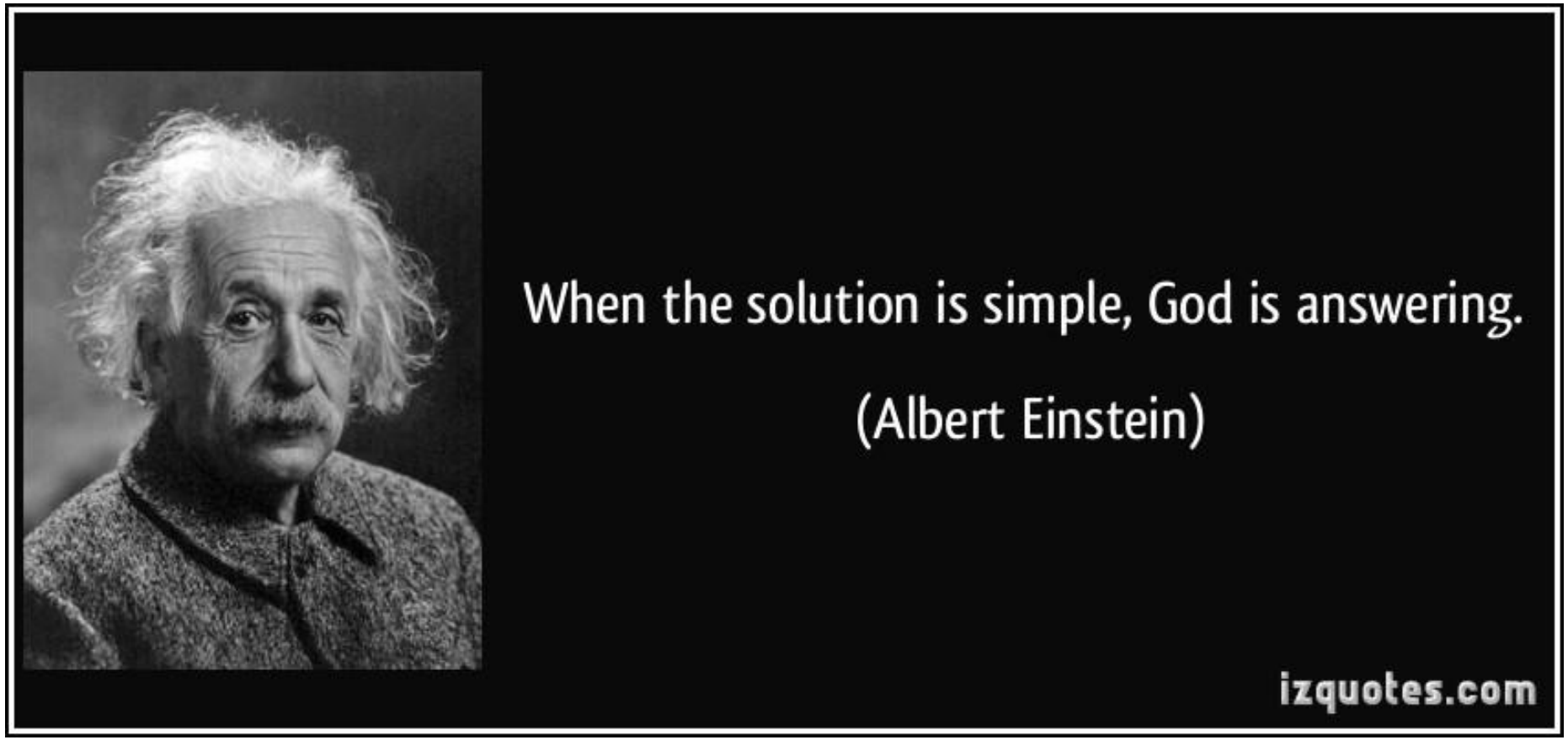
- 替代主流的PLDA，不需要指定子空间的维数且提高了说话人识别的性能。

2. Angular Softmax

- 不需要选取Triplet数据选择，更容易使用且训练稳定。

3. CTC-CRF

- 在WSJ、Switchboard、Librispeech，性能表现均超过了CTC、Attention Seq2Seq以及现在广为流行的Kaldi工具包中的端对端模型（End-to-end Chain-model）；
- 同时具有训练流程简洁、能充分利用词典及语言模型从而数据利用效率高等优势。



Theoretical Sound, Simple, Scalable.

Thanks for your attention !

Thanks to my students :

Yutian Li, Yiyan Wang, Hongyu Xiang.