Remarks on Optimal Scores for Speaker Recognition

Dong Wang

2020/11/21

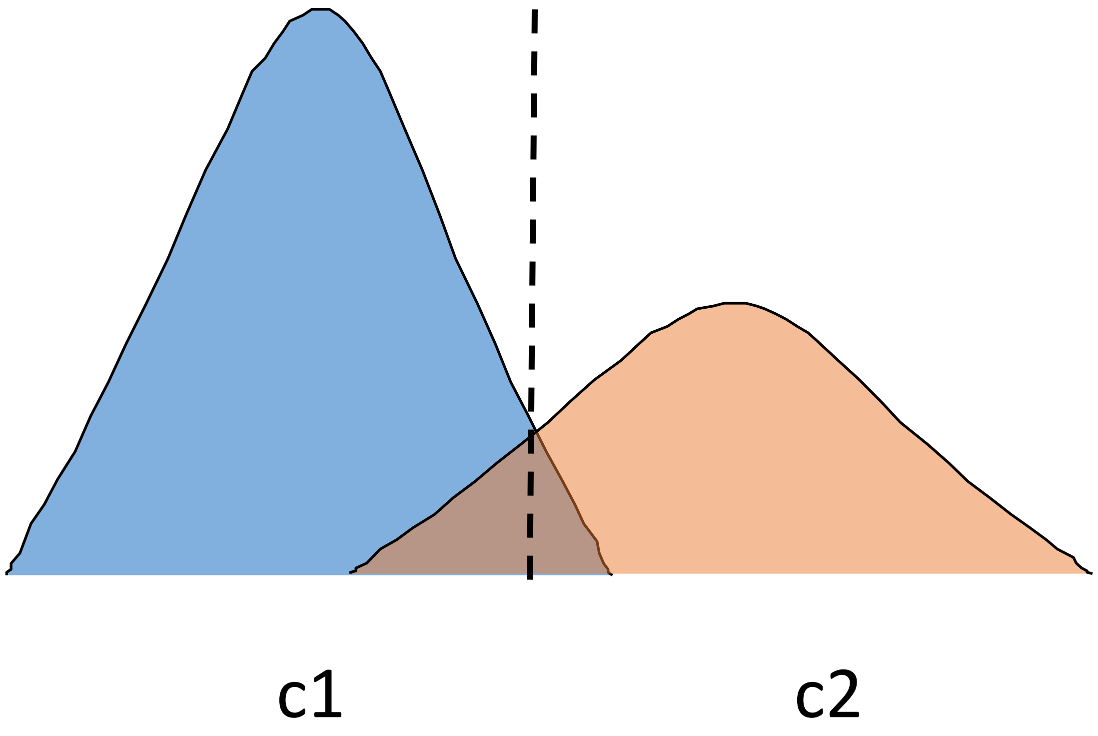
     

What is optimal score?

- Answer: they should lead to minimum Bayes risk
- Maximum a Posterior (MAP) principle

$$c^* = \text{armgax}_c\ p(c|x)$$

- For speaker identification, it is simple



Optimal score for verification

- Two-class problem
  - H0: spoken by speaker k
  - H1: not spoken by speaker k
- MAP principle
  - p(H0|x) = p(x|H0) /(p(x|H0) + p(x|H1))
- Only the likelihood ratio (LR) matters:

$$\frac{p(\boldsymbol{x}|H_0)}{p(\boldsymbol{x}|H_1)} = \frac{p_k(\boldsymbol{x})}{p(\boldsymbol{x})}$$

- It is widely used in GMM-UBM era, but derived from hypothesis test.

Dong Wang, "Remakrs on optimal scores for speaker recognition", 2020, http://arxiv.org/abs/2010.04862
Dong Wang, "A Simulation Study on Optimal Scores for Speaker Recognition", EURASIP Journal on Audio, Speech, and Music Processing, 2020.

Normalized Likelihood

- We call the likelihood ratio $p_k(x)/p(x)$ Normalized Likelihood

$$NL(\boldsymbol{x}|k) = \frac{p(\boldsymbol{x}|H_0)}{p(\boldsymbol{x}|H_1)} = \frac{p_k(\boldsymbol{x})}{p(\boldsymbol{x})}$$

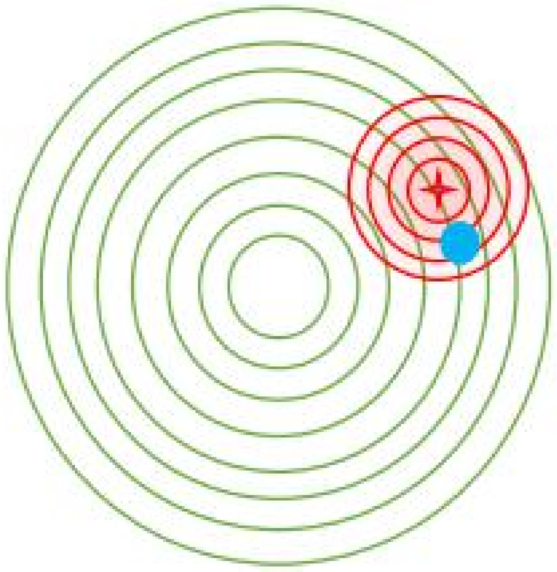
- It is a speaker-dependent likelihood normalized by a speaker-independent likelihood
- It is a special LR, different from other forms, e.g., the LR in PLDA, i.e., p(x,y)/p(x)p(y)
- It is the simple, general form that leads to MBR decision.

We employ NL to scoring embeddings (back-end modeling)

- Suppose both prior p(u) and condition p(x|u) are Gaussians

$$p(\boldsymbol{\mu}) = N(\boldsymbol{\mu}; \mathbf{0}, \mathbf{I}\epsilon^2)$$

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = N(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2\mathbf{I})$$



- We can compute H1

$$p(\boldsymbol{x}) = N(\boldsymbol{x}; \mathbf{0}, \mathbf{I}(\boldsymbol{\epsilon}^2 + \sigma^2))$$

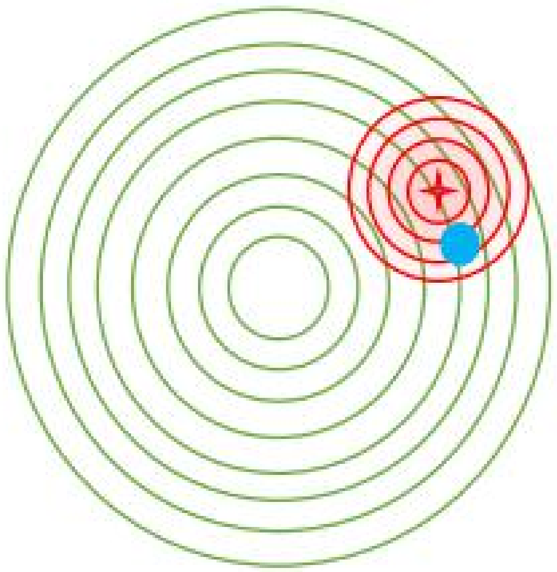
Now compuet H0

- Suppose we enroll speaker k using $x_1^k, \ldots, x_{nk}^k$, and need compute H1.

$$p_k(\boldsymbol{x}) = N(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})$$

- Compute $p(u|x_1^k, \ldots, x_{nk}^k)$.

$$p(\boldsymbol{\mu}_k|\boldsymbol{x}_1^k, \ldots \boldsymbol{x}_{n_k}^k) = N(\boldsymbol{\mu}_k; \frac{n_k \boldsymbol{\epsilon}^2}{n_k \boldsymbol{\epsilon}^2 + \sigma^2} \bar{\boldsymbol{x}}_k, \mathbf{I} \frac{\sigma \boldsymbol{\epsilon}^2}{n_k \boldsymbol{\epsilon}^2 + \sigma^2})$$



Compuet NL

- Now marginalize over u:

$$\begin{aligned} p_k(\boldsymbol{x}) &= p(\boldsymbol{x}|\boldsymbol{x}_1^k, ..., \boldsymbol{x}_{n_k}^k) \\ &= \int p(\boldsymbol{x}|\boldsymbol{\mu}_k)p(\boldsymbol{\mu}_k|\boldsymbol{x}_1^k, ...\boldsymbol{x}_{n_k}^k)\mathrm{d}\boldsymbol{\mu}_\mathrm{k} \\ &= N(\boldsymbol{x}; \frac{n_k\boldsymbol{\epsilon}^2}{n_k\boldsymbol{\epsilon}^2+\sigma^2}\bar{\boldsymbol{x}}_k, (\sigma^2 + \mathbf{I}\frac{\sigma\boldsymbol{\epsilon}^2}{n_k\boldsymbol{\epsilon}^2+\sigma^2})) \end{aligned}$$

- NL score obtained:

$$NL(\boldsymbol{x}|k) = \frac{p(\boldsymbol{x}|\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_k})}{p(\boldsymbol{x})}$$

$$\log NL(\boldsymbol{x}|k) \propto -||\frac{\boldsymbol{x}-\tilde{\boldsymbol{\mu}}_k}{\sqrt{\sigma^2+\frac{\boldsymbol{\epsilon}^2\sigma^2}{n_k\boldsymbol{\epsilon}^2+\sigma^2}}}||^2 + ||\frac{\boldsymbol{x}}{\sqrt{\boldsymbol{\epsilon}^2+\sigma^2}}||^2$$

Remark 1: It equals to PLDA with linear Gaussian

- PLDA score is a likelihood ratio in a different form

$$LR_{PLDA}(\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{x}_1, ..., \boldsymbol{x}_n)}{p(\boldsymbol{x})p(\boldsymbol{x}_1, ..., \boldsymbol{x}_n)}$$

- But they are the same!

$$LR_{PLDA}(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)}{p(\boldsymbol{x})} = \frac{\int p(\boldsymbol{x}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)\mathrm{d}\boldsymbol{\mu}}{p(\boldsymbol{x})}$$

Remark1: It equals to PLDA with linear Gaussian (2)

- What is new?
  - NL computes the score in a more efficient way
  - NL divides the scoring into three steps: enroll, prediction, normalization

$$LR_{PLDA}(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)}{p(\boldsymbol{x})} = \frac{\int p(\boldsymbol{x}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)\mathrm{d}\boldsymbol{\mu}}{p(\boldsymbol{x})}$$

  - NL allows separate models for H0 and H1.
- Anyway, all the properties that we will discuss are shared by PLDA.

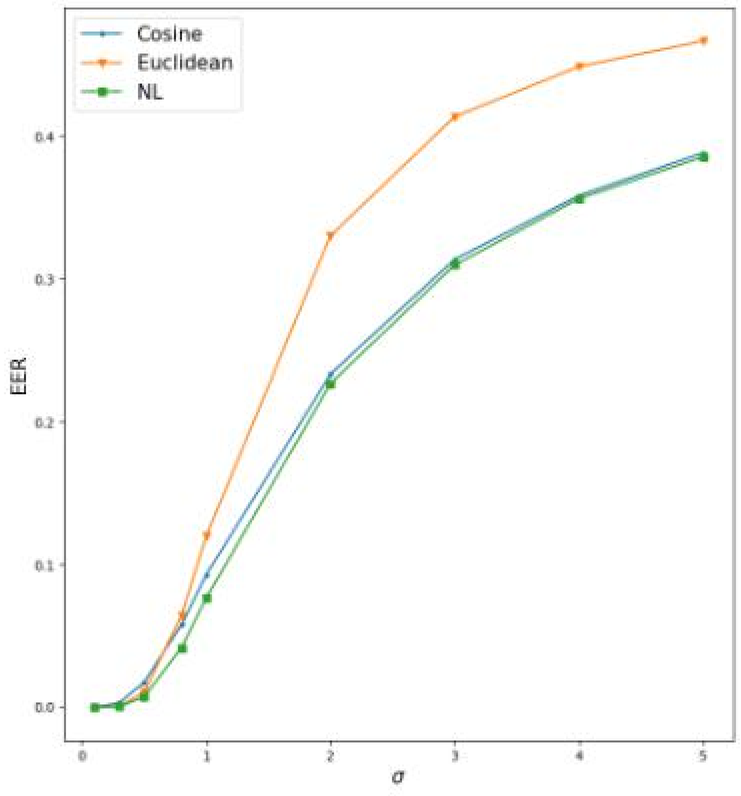
Remark 2: Cosine and Euclidean score are appxomiation of NL

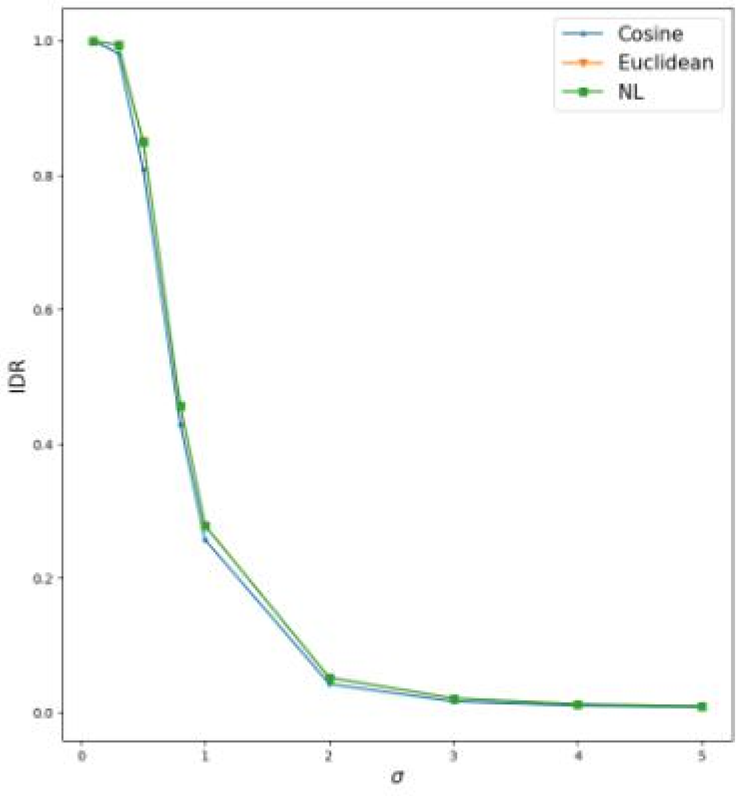
- Reformulate NL

$$\log NL(\boldsymbol{x}|k) \propto -\Big\{\frac{n_k\boldsymbol{\epsilon}^4}{(\sigma^2+\boldsymbol{\epsilon}^2)(n_k\boldsymbol{\epsilon}^2+\sigma^2)}||\boldsymbol{x}||^2 + ||\tilde{\boldsymbol{\mu}}_k||^2 - 2\cos(\boldsymbol{x},\tilde{\boldsymbol{\mu}}_k)||\boldsymbol{x}||\;||\tilde{\boldsymbol{\mu}}_k||\Big\}$$

- When ε is large, it converts to Euclidean score
- When σ is large, it converts to Cosine score

Simulation





Remark 3: NL score is optimal for both SV and SI

- The only difference is in the normalization
- We never need to consider different scores for different tasks.

$$NL(\boldsymbol{x}|k) = \frac{p(\boldsymbol{x}|H_0)}{p(\boldsymbol{x}|H_1)} = \frac{p_k(\boldsymbol{x})}{p(\boldsymbol{x})}$$

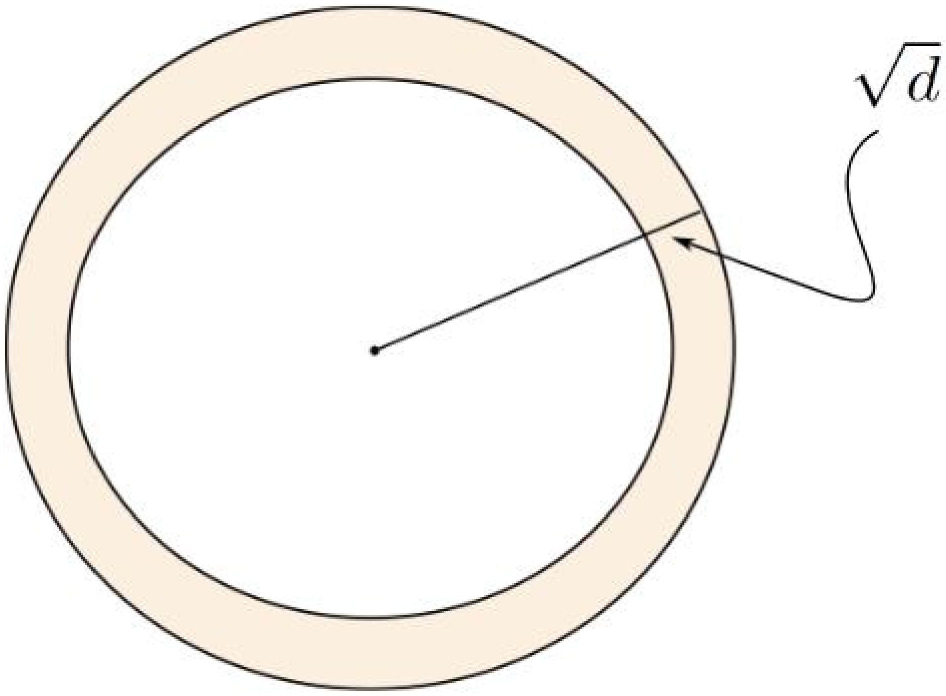
Remark 4: NL is invariant to any invertible transform

- Any invertible transform will lead to the same NL score
- It is a very important property that allows us to perform distribution manipulation

$$
\begin{aligned}
NL(g(\boldsymbol{x})|g(\boldsymbol{x}_1),...,g(\boldsymbol{x}_{n_k})) &= \frac{p'(g(\boldsymbol{x}),g(\boldsymbol{x}_1),...,g(\boldsymbol{x}_{n_k}))}{p'(g(\boldsymbol{x}))p'(g(\boldsymbol{x}_1),...,g(\boldsymbol{x}_{n_k}))} \\
&= \frac{J(\boldsymbol{x})\prod_{i=1}^{n} J(\boldsymbol{x}_i)p(\boldsymbol{x},\boldsymbol{x}_1,...,\boldsymbol{x}_{n_k})}{\{J(\boldsymbol{x})p(\boldsymbol{x})\}\{\prod_{i=1}^{n} J(\boldsymbol{x}_i)p(\boldsymbol{x}_1,...,\boldsymbol{x}_{n_k})\}} \\
&= NL(\boldsymbol{x}|\boldsymbol{x}_1,...,\boldsymbol{x}_{n_k})
\end{aligned}
$$

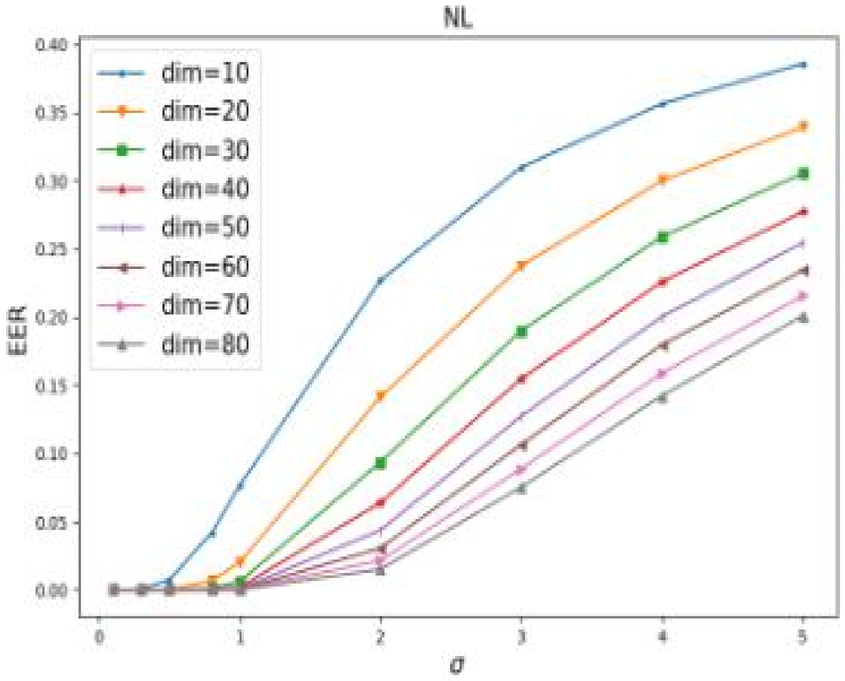
Remark 5: dimension is important

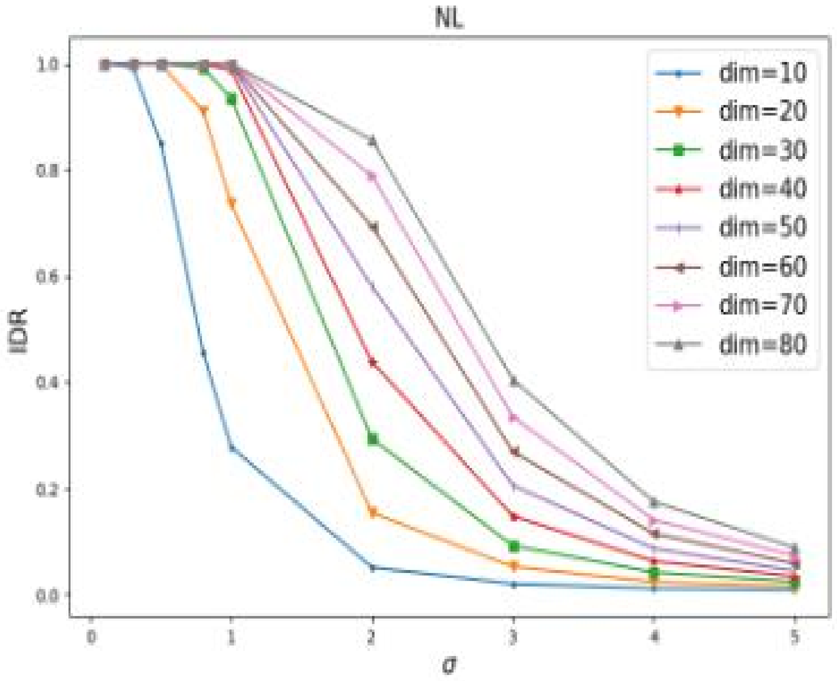
- Gaussian annulus theorem: nearly all the high-dimensional Gaussian vectors concentrate on a thin spherical surface.
- Length-norm employs this property.



Remark 5: dimension is important(2)

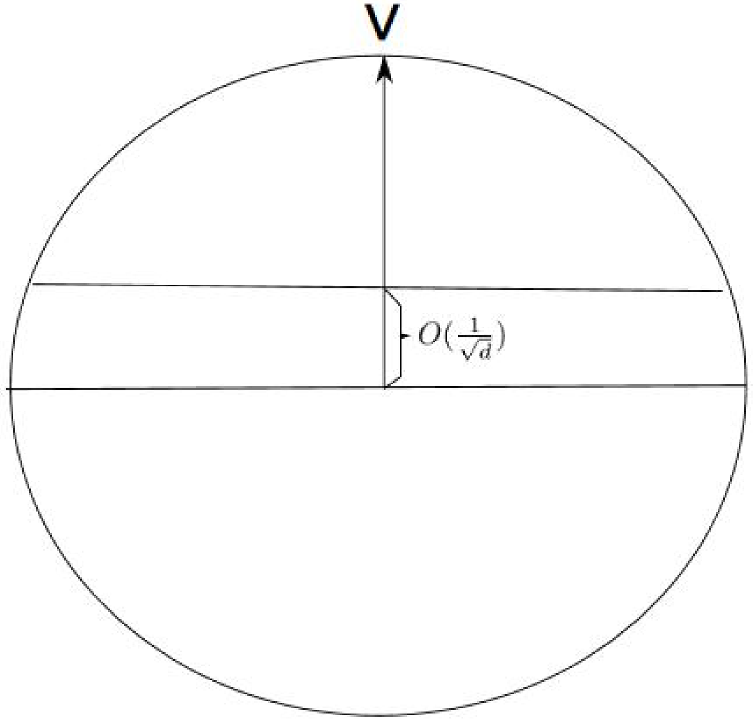
- More dimensions lead to better discrimination
- If $\sigma^2 < O(\varepsilon^4 d)$, any two vectors tend to be separated





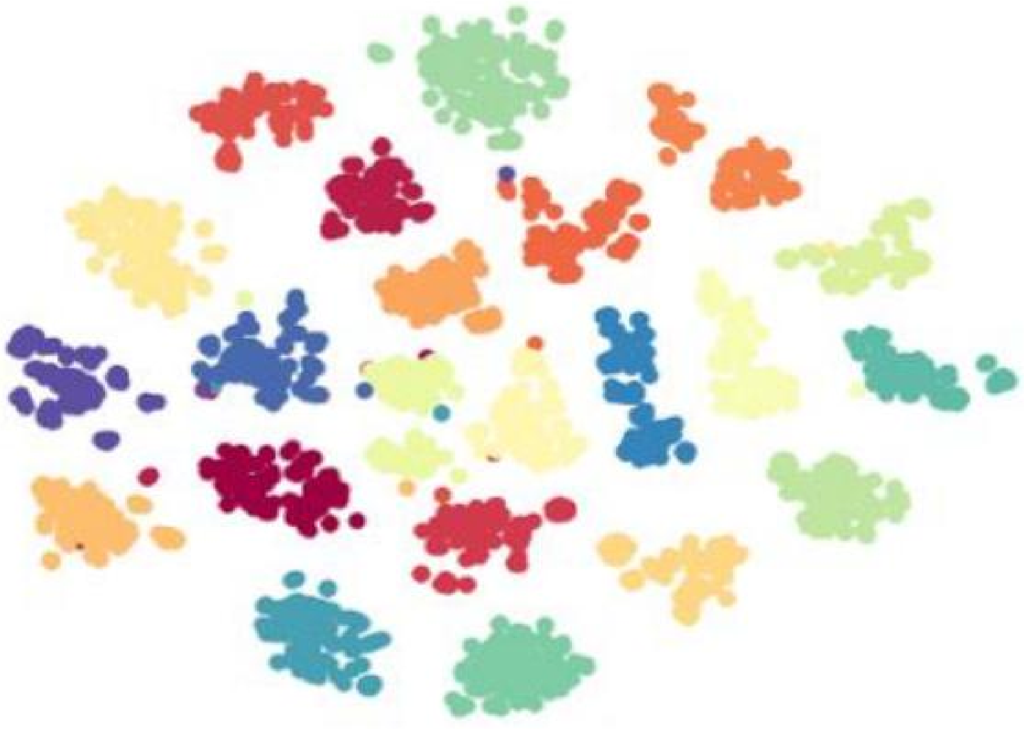
Remark 6: Direction is imporant

- For any vector x as a pole, most other vector concentrate on the equator
- Most of the vectors are orthogonal



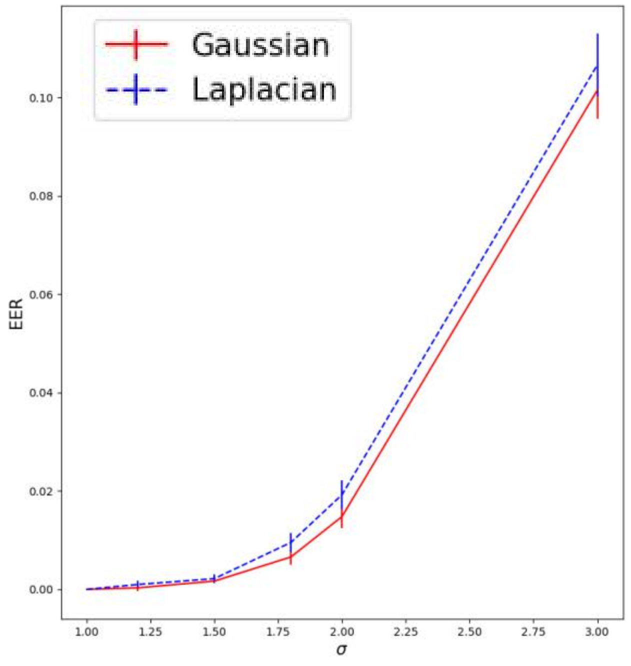
All seem interesting, but...

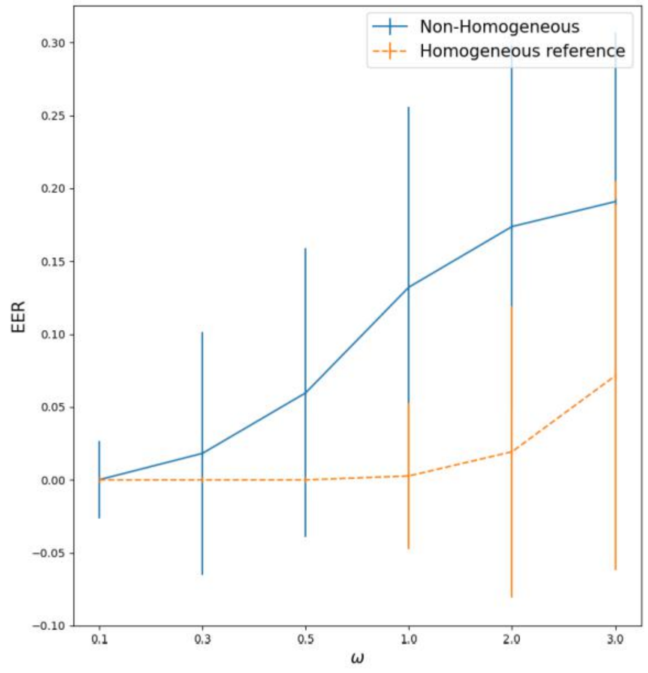
- Almost all the remarks are based on the linear Gaussian assumption
- If the vectors are, we get optimal decisions, but are they?



Consequence of incorrect distributions

- Non-Gaussianality and Non-homogeneity corrupt NL



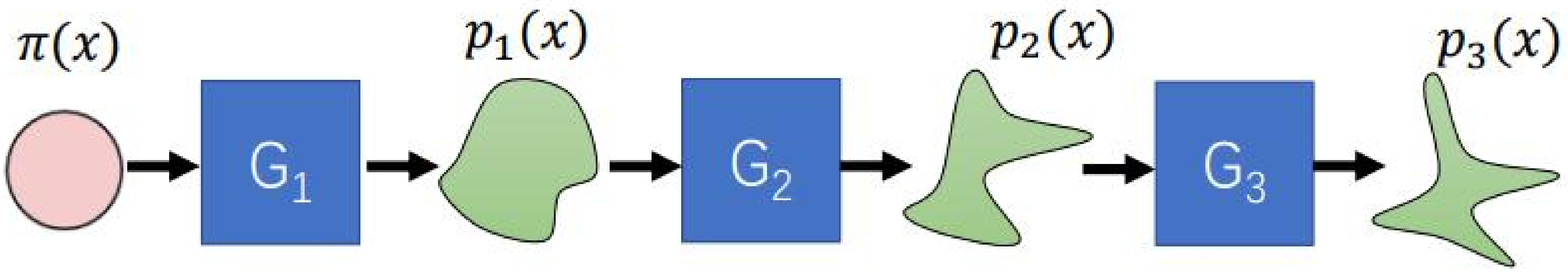


Dong Wang, "A Simulation Study on Optimal Scores for Speaker Recognition", EURASIP Journal on Audio, Speech, and Music Processing, 2020.

- Transform Non-Gaussian to Gaussian
- Transform Non-homogeneous to homogeneous

Deep normalization: make distributions Gaussian

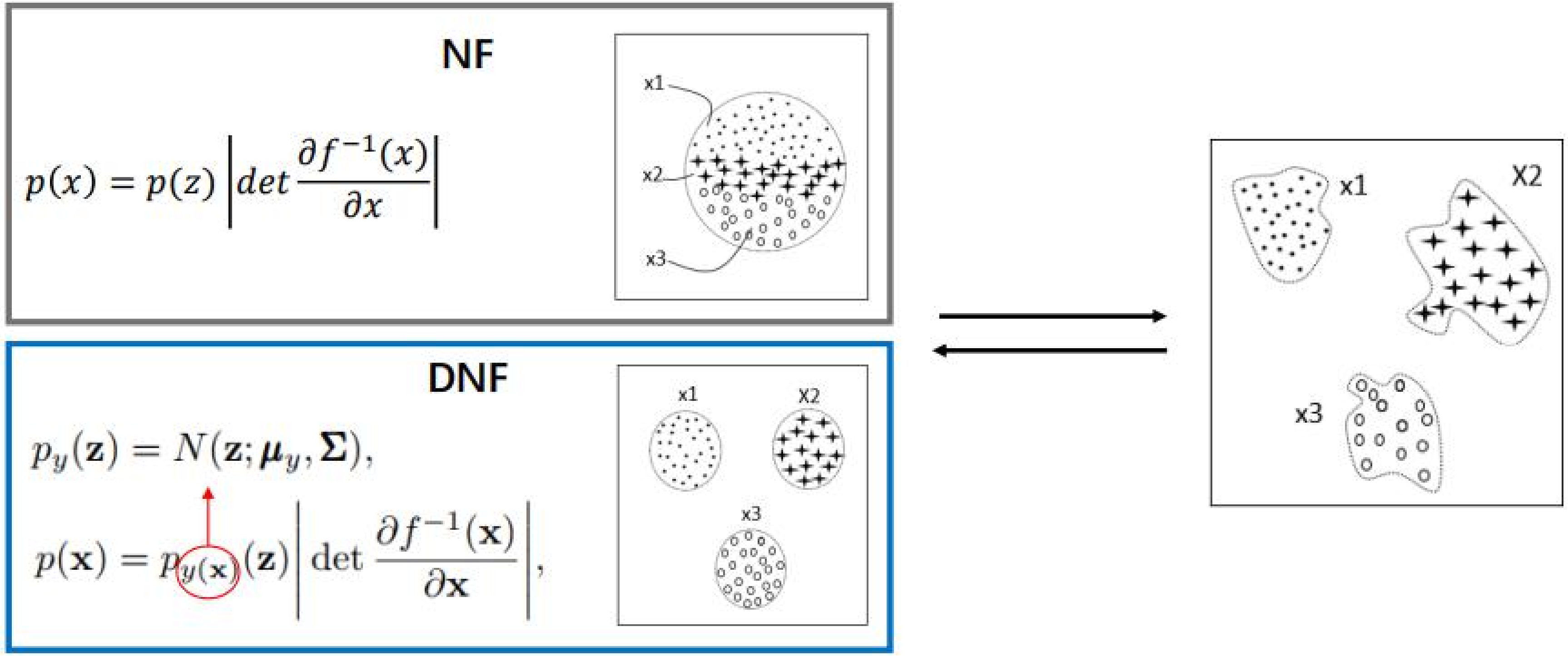
- Transform to non-Gaussian to Gaussian



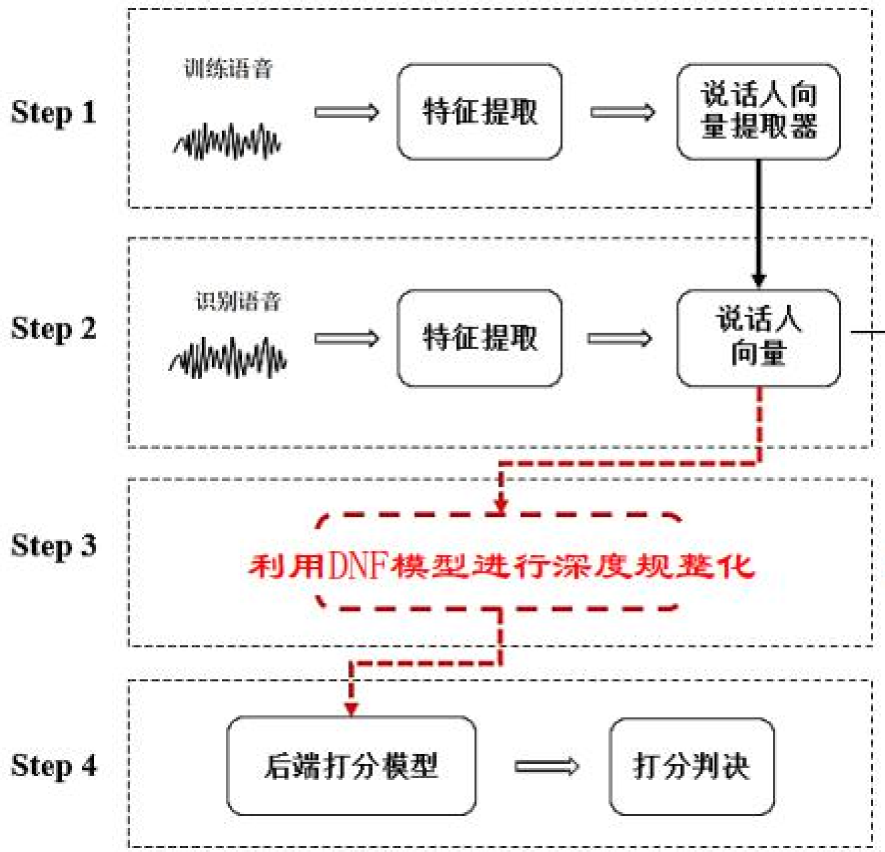
$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \log|\det(d\mathbf{z}/d\mathbf{x})| \\ &= \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \boxed{\sum_{i=1}^{K} \log|\det(d\mathbf{h}_i/d\mathbf{h}_{i-1})|}\end{aligned}$$

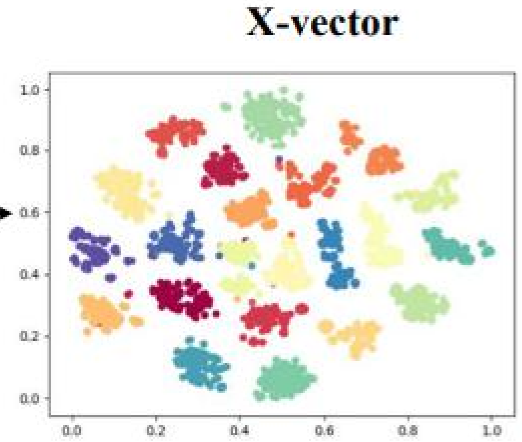
Deep normalization (2)

- Try to make each class Gaussian



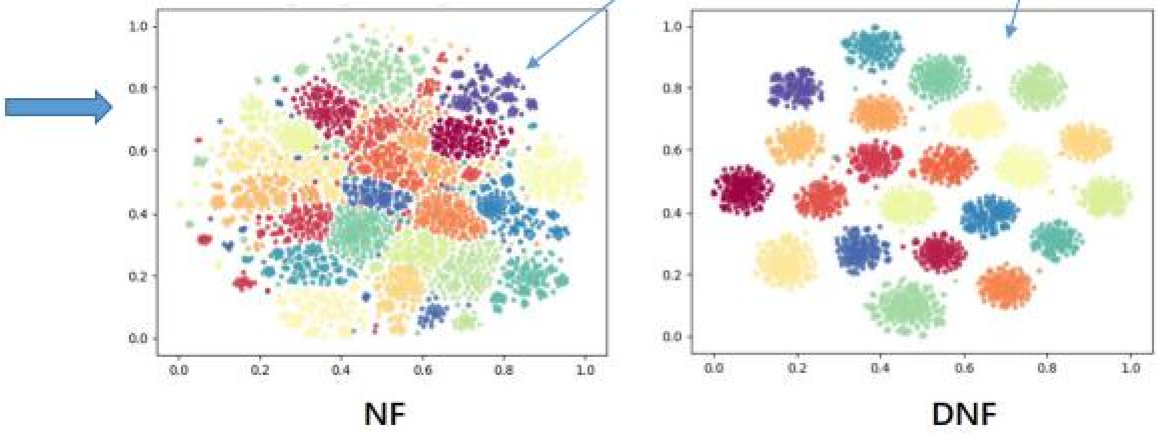
Deep normalization (2)





Xvector,及其规整化

对比了NF模型和DNF模型
的规整化效果



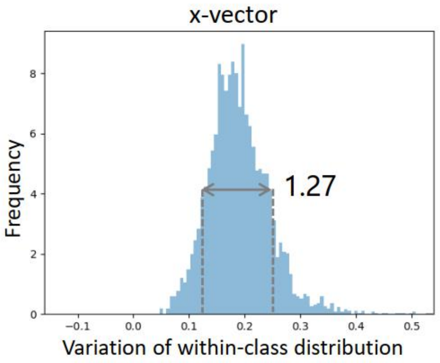
Deep normalization (3)

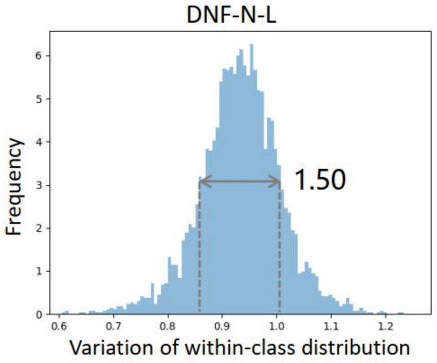
		SITW		CNCeleb	
		Cosine	PLDA	Cosine	PLDA
TDNN	x-vector [512]	17.20	5.30	16.32	13.03
	LDA [200]	5.82	3.96	17.52	13.50
	DNF [512]	8.53	3.66	14.22	11.82
	DNF-LDA [200]	5.41	3.42	15.18	13.22
TDNN + Att.	x-vector [512]	4.37	3.66	15.08	13.05
	LDA [200]	3.72	2.73	18.34	13.97
	DNF [512]	5.00	2.71	14.69	12.07
	DNF-LDA [200]	3.72	2.57	15.45	13.66
ResNet-34 + Att.	x-vector [512]	2.73	2.52	13.94	13.11
	LDA [200]	2.60	2.00	14.90	12.58
	DNF [512]	3.47	1.94	**13.86**	**11.61**
	DNF-LDA [200]	**2.57**	1.89	14.04	12.32
ResNet-34 + AAM	x-vector [512]	5.71	2.82	15.80	14.02
	LDA [200]	2.73	1.86	16.67	13.42
	DNF [512]	4.89	2.32	14.66	12.80
	DNF-LDA [200]	2.93	**1.83**	14.96	12.59

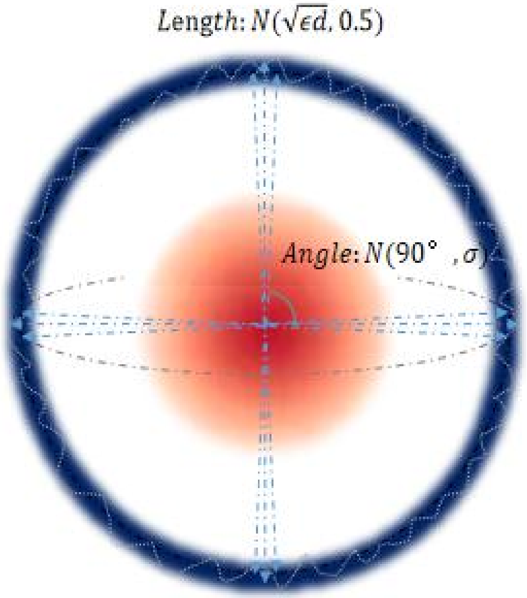
- Yunqi Cai, Lantian Li, Andrew Abel, Xiaoyan Zhu, Dong Wang, Deep normalization for speaker vectors, IEEE TASLP 2020.

Maximum Gaussian training: Make distributions homogeneous

- Vanilla deep norm by ML cannot ensure homogeneous
- Train to maximizing Gaussian, according to Remark 5 and 6.



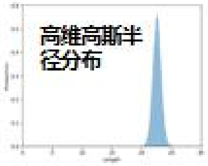




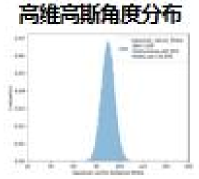
Length metric:

$$p(\ell(\mathbf{x})) = N(\sqrt{\epsilon d}, 0.5\epsilon).$$

$$\mathcal{R}_\ell = -\sum_i ||\ell(\mathbf{x}_i) - \sqrt{\epsilon d}||^2$$



Angle metric:

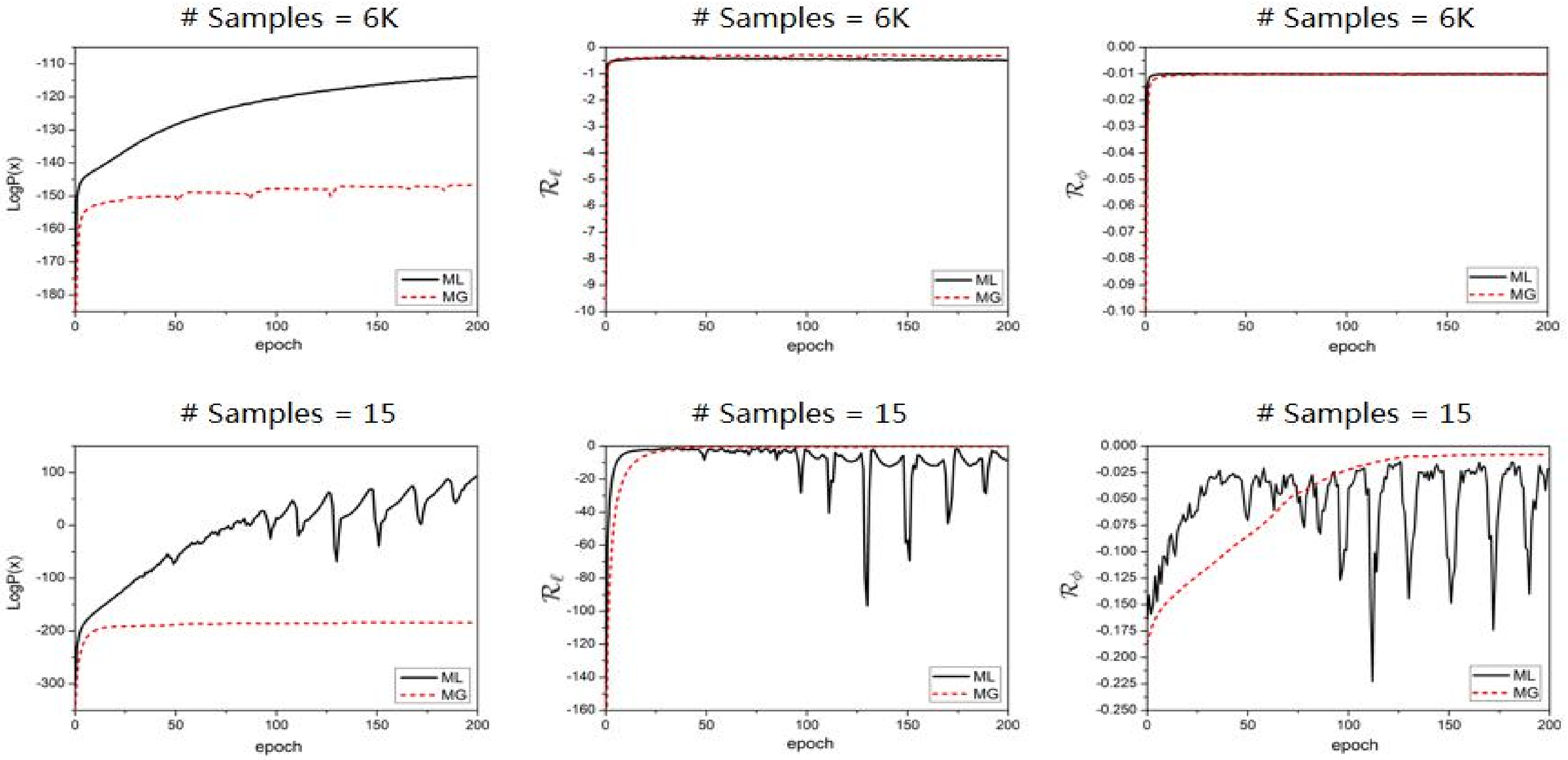


$$p(\omega(\mathbf{x}_1, \mathbf{x}_2)) = N(90°, \sigma),$$

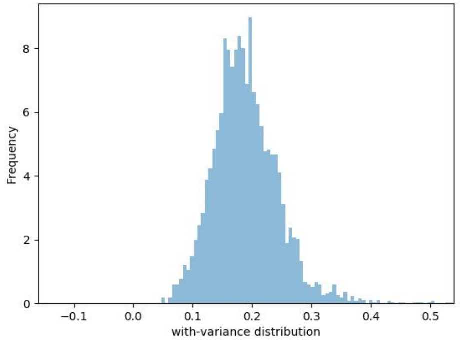
$$\mathcal{R}_\phi = -\sum_i \sum_j \frac{||\phi(\mathbf{x}_i, \mathbf{x}_j)||^2}{2\xi},$$

Maximum Gaussian training (2)

- MG training is much more stable



Maximum Gaussian training (3)



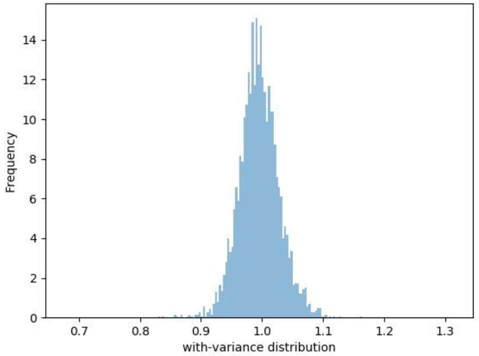


TABLE III: EER(%) results on SITW and CNCeleb with DNF variants.

Models	Between-Class Criterion	Within-Class Criterion	SITW		CNCeleb	
			Cosine	PLDA	Cosine	PLDA
x-vector	N/A	N/A	17.20	5.30	16.32	13.03
DNF-N-L	N/A	ML	8.53	3.66	14.22	11.82
DNF-L-L	ML	ML	10.47	3.72	15.83	**11.39**
DNF-G-G	MG	MG	**6.30**	**3.37**	**12.13**	11.72
DNF-G-L	MG	ML	6.89	3.45	13.99	11.46
DNF-G-LG	MG	ML+MG	6.42	3.36	12.96	11.51

- Yunqi Cai, Lantian Li, Andrew Abel, Xiaoyan Zhu, Dong Wang, “Maximum Gaussian training for speaker normalization”,https://arxiv.org/abs/2010.16148

Further remarks

- Is the deepnorm really optimal?
  - According remark 4, any invertible transform does not change the NL score
  - The NL score in the latent space is as optimal as in the observation space, however the data is more Gaussian and so amiable for NL modeling.
- With deep norm, it seems we don’t need try to derive other powerful scores and score calibration...
- Most research focus on discrimination, though normalization should be emphasized.

A case study: Tackle the enroll-test conditional mismatch

- We treat the conditional mismatch as a problem of mismatch on statistics
- NL provides an elegent framework for deal with ‘decoupled’ computation on mismatched conditions

Enroll

Prediction

$$\frac{\int p(\boldsymbol{x}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)\mathrm{d}\boldsymbol{\mu}}{p(\boldsymbol{x})}$$

Normalization

A case study: Tackle the enroll-test conditional mismatch (2)

Cases	Base	Methods		
		MDT	DAT	DSD
AND-AND	0.797	-	-	-
AND-Mic	2.146	1.151	1.245	**0.981**
AND-iOS	1.425	1.161	1.312	**0.623**
Mic-AND	2.175	1.161	1.189	**0.712**
Mic-Mic	0.778	-	-	-
Mic-iOS	2.251	1.293	1.481	**0.812**
iOS-AND	1.599	1.156	1.184	**0.755**
iOS-Mic	2.216	1.137	1.231	**1.052**
iOS-iOS	0.920	-	-	-

- Lantian Li, Yang Zhang, Jiawen Kang, Thomas Fang Zheng, Dong Wang, SQUEEZING VALUE OF CROSS-DOMAIN LABELS: A DECOUPLED SCORING APPROACH FOR SPEAKER VERIFICATION, submitted to ICASSP 2021.

Conclusions

- Normalization likelihood is the optimal score for both SV and SI, in terms of **minimum Bayes risk**. It is equal to the PLDA likelihood ratio, but with clear advantage.
- NL requires regularized distributions. **Deep normalization** can do that.
- NL brings many interesting things: **decoupling, interpretation, nonlinear model**…
- Finally, NL provides **a 'bound'** of the performance, which seems an advantage of the embedding approach, when compared to end-to-end methods.

Reference papers

- Dong Wang, "***Remakrs on optimal scores for speaker recognition***", 2020, http://arxiv.org/abs/2010.04862
- Dong Wang, "***A Simulation Study on Optimal Scores for Speaker Recognition***", EURASIP Journal on Audio, Speech, and Music Processing, 2020. https://arxiv.org/pdf/2004.04095.pdf
- Yunqi Cai, Lantian Li, Andrew Abel, Xiaoyan Zhu, Dong Wang, "***Maximum Gaussian training for speaker normalization***",https://arxiv.org/abs/2010.16148
- Yunqi Cai, Lantian Li, Andrew Abel, Xiaoyan Zhu, Dong Wang, ***Deep normalization for speaker vectors***, IEEE TASLP 2020. https://arxiv.org/pdf/2004.04095
- Lantian Li, Dong Wang, Thomas Fang Zhang, ***Neural Discriminant Analysis for Deep Speaker Embedding***, Interspeech 2020. https://arxiv.org/pdf/2005.11905
- Lantian Li, Yang Zhang, Jiawen Kang, Thomas Fang Zheng, Dong Wang, ***SQUEEZING VALUE OF CROSS-DOMAIN LABELS: A DECOUPLED SCORING APPROACH FOR SPEAKER VERIFICATION***, submitted to ICASSP 2021. https://arxiv.org/pdf/2010.14243

Welcome join us!