Statistics decomposition for NL Scoring (SD-NL)

Di Wang

2020-11-17

What is NL?

- Normalized likelihood
  - $p(x|H_0)$: denotes as $p_c(x)$,which is a speaker-dependent item.
  - $p(x|H_1)$: denotes as $p(x)$,which is a speaker-independent item.

$$NL(x|c) = \frac{p(x|H_0)}{p(x|H_1)} = \frac{p_c(x)}{p(x)}$$

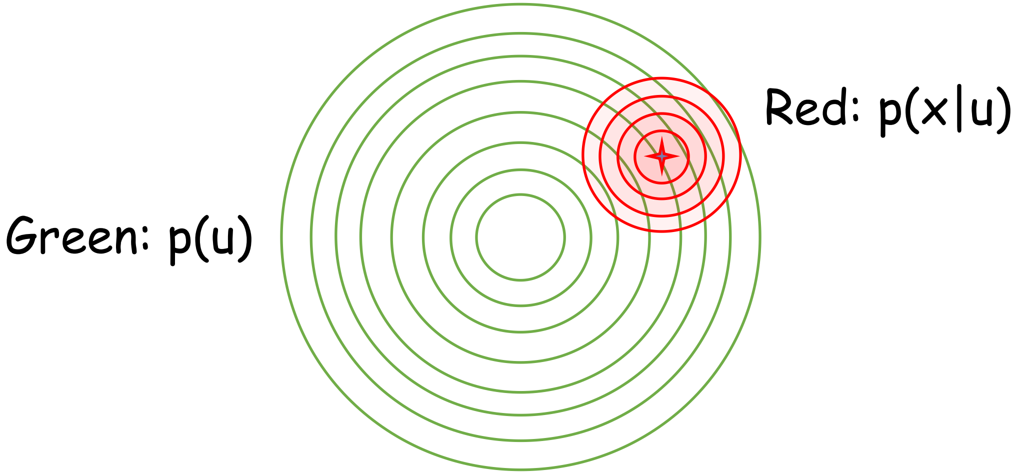
NL model

- A linear Gaussian model

$$p(u) = N(u;0,\varepsilon I)$$
$$p(x \mid u) = N(x;u,\sigma I)$$

$$p(x) = \int p(x \mid u)\,p(u)\,du$$
$$= N(x;0,(\varepsilon+\sigma)I)$$

$$NL(x \mid u_c) = \frac{p(x \mid u_c)}{p(x)}$$
$$= \frac{p_c(x)}{p(x)}$$
$$= \frac{p(x \mid x_1^c, x_2^c, \ldots, x_n^c)}{p(x)}$$



$$p(x) = \int p(x \mid u)\,p(u)\,du$$

Three components of NL

$$NL(x|u_c)=\frac{p(x|u_c)}{p(x)}=\frac{p_c(x)}{p(x)}=\frac{p(x|x_1^c,\dots x_n^c)}{p(x)}=\frac{\int p(x|u)p(u|x_1^c,\dots x_n^c)du}{\int p(x|u)p(u)du}$$

I can predict unknown classes~~haha

- Decouple NL to **three** components
  - Enrollment : $p(u|x_1^c,\dots x_n^c)$ produces the posterior of class mean.
  - Prediction : $p(x|u)$ computes the likelihood of x belonging to class c.
  - Normalization : $p(x)$ computes the likelihood of x from all classes.

Why we need decouple ?

- Background :
  - In practice, the data could be quite complex.
  - NL/PLDA is modeled by between-var and within-var, and it uses the same statistics for different components, which is obviously unreasonable.

- Ideal:
  - The different components use their own optimal model.

- So we need decouple,and :
  - A high-level and global perspective that overlooks the distributional of the entire data.
  - A low-level and local perspective that scrutinizes the distribution of a single class.

How to implement decouple ?

- Enrollment $p(u \mid x_1^c, \ldots x_n^c)$ and Normalization $p(x)$ are relevant to a global generative model, e.g., PLDA.
  - $p_g(u) = N(u; 0, \varepsilon I)$
  - $p_g(x \mid u) = N(x; u, \sigma I)$

- Predication $p(x \mid u_c)$ regards as a local model
  - $p_l(x \mid u) = N(x; u, \sum')$

$$NL(x \mid u_c) = \frac{p(x \mid u_c)}{p(x)} = \frac{p_c(x)}{p(x)} = \frac{\int p_l(x \mid u) p_g(u \mid x_1^c, \ldots x_n^c) du}{\int p_g(x \mid u) p_g(u) du}$$

Training process - Global

- Global training
  - ML-PLDA

$$p(x_1,...x_n) \propto |\sigma I|^{-\frac{n}{2}} |\varepsilon I|^{-\frac{1}{2}} \left| (\frac{n}{\sigma} + \frac{1}{\varepsilon}) I \right|^{-\frac{1}{2}}$$

$$\exp\left\{ -\frac{1}{2\sigma} \left\{ \sum_i \|x_i\|^2 - \frac{n^2 \varepsilon}{n\varepsilon + \sigma} \|\bar{x}\|^2 \right\} \right\}$$

where | · | defined is the absolute value of the determinant of a matrix. Given a training set consisting of K classes, the parameters ε and σ can be optimized by maximizing the likelihood function:

$$L(\varepsilon, \sigma) = \sum_{c=1}^{C} p(x_1^c, ..., x_{n_c}^c)$$

where x_i^c is the i-th sample of the c-th class.

◆ Inference :

$$Given X = \{x_1, x_2, ..., x_n\}$$

$$\begin{aligned} p(X) &= p(x_1, x_2, ..., x_n) \\ &= \int p(x_1, x_2, ..., x_n \mid u) p(u) du \\ &= \int p(x_1 \mid u) p(x_2 \mid u) ... p(x_n \mid u) p(u) du \end{aligned}$$

$$\propto |\sigma I|^{-\frac{n}{2}} |\varepsilon I|^{-\frac{1}{2}} \left| (\frac{n}{\sigma} + \frac{1}{\varepsilon}) I \right|^{-\frac{1}{2}}$$

$$\exp\left\{ -\frac{1}{2\sigma} \left\{ \sum_i \|x_i\|^2 - \frac{n^2 \varepsilon}{n\varepsilon + \sigma} \|\bar{x}\|^2 \right\} \right\}$$

Training process - Local

- Local training
  - MLLR $x^{'} = Mx$

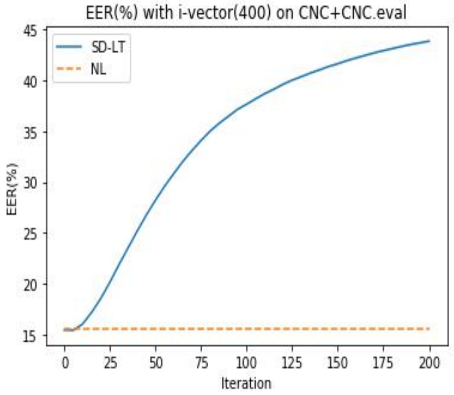
$$L(M) = \prod_{c}^{C} \prod_{i=1}^{n_c} \int p_l(x_i^c \mid u, \sum^{'}) p_g(u \mid x_1^c, \ldots, x_{n_c}^c) du$$

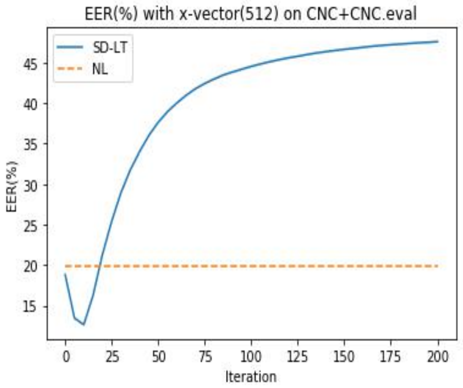
$$= \prod_{c}^{C} \prod_{i=1}^{n_c} \int p_g(Mx_i^c \mid u, \sigma I) p_g(u \mid x_1^c, \ldots, x_{n_c}^c) du$$

$$= \prod_{c}^{C} \prod_{i=1}^{n_c} N(Mx_i^c ; \frac{n_k \varepsilon}{n_k \varepsilon + \sigma} \bar{x}_k, I(\sigma + \frac{\varepsilon \sigma}{n_k \varepsilon + \sigma}))$$

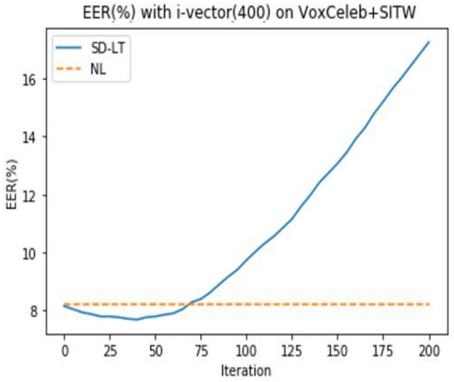
Basic results

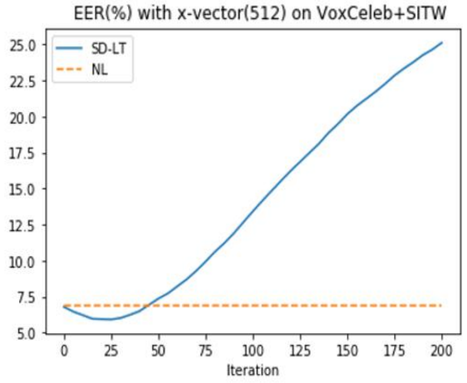
- Results on CNC.eval





- Results on SITW





EER(%) results on CNC.eval

	SD-LT（best）	NL
i-vector	15.44%	15.56%
x-vector	12.63%	19.79%

EER(%) results on SITW

	SD-LT（best）	NL
i-vector	15.44%	15.56%
x-vector	12.63%	19.79%

Statistics analysis

- Changes in kurtosis and skewness during training of iVector and XVector on CNC (dev set and test set).

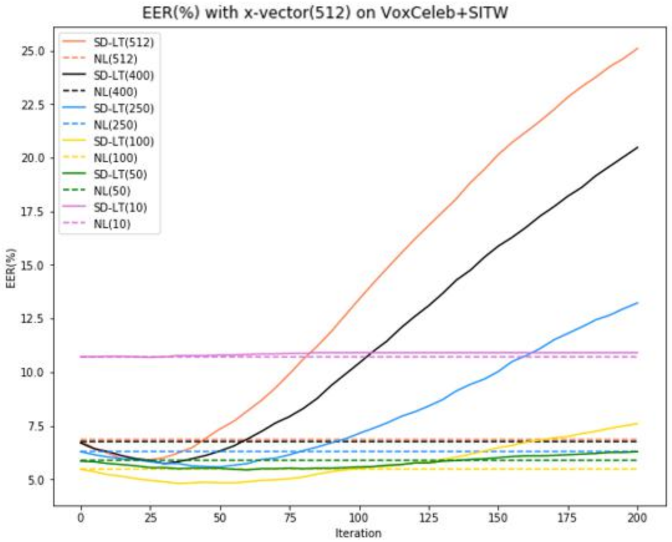
dev set	skew	kurt
iVector	0.001239	0.303497
xVector	-0.001600	0.300418

test set	skew	kurt
iVector	0.003140	0.266604
xVector	0.004902	1.513297

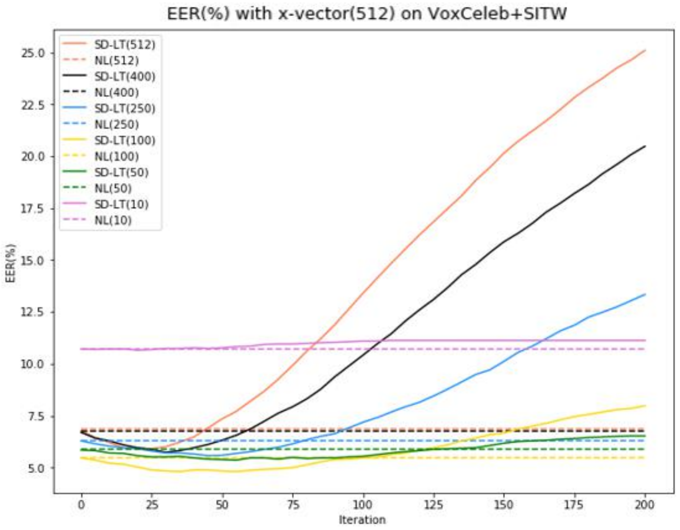
Dimensional reduction

- The data is dimensionally reduced in two different ways and the performance in different dimensions is observed.

Dimension reduction in grading



Dimension reduction in transforming



	Dims (best/basic)	EER
i-vector on SITW	100/400	7.11%/7.68%
x-vector on SITW	100/512	4.81%/5.91%
i-vector on CNC.eval	250/400	15.39%/15.44%
x-vector on CNC.eval	250/512	12.15%/12.63%

Relation to Length-norm(LN)

- The relationship between NL, SD/LT and standard LN was compared.

EER(%) with x-vector(512) on CNC.eval

	Optimal results(EER)
Basic NL	19. 79%
+LN	12. 71%
+SD/LT	<u>12. 63%</u>

EER(%) with x-vector(512) on SITW

	Optimal results(EER)
Basic NL	6. 86%
+LN	<u>4. 51%</u>
+SD/LT	5. 91%

EER(%) with i-vector(400) on CNC.eval

	Optimal results(EER)
Basic NL	15. 56%
+LN	15. 69%
+SD/LT	<u>15. 44%</u>

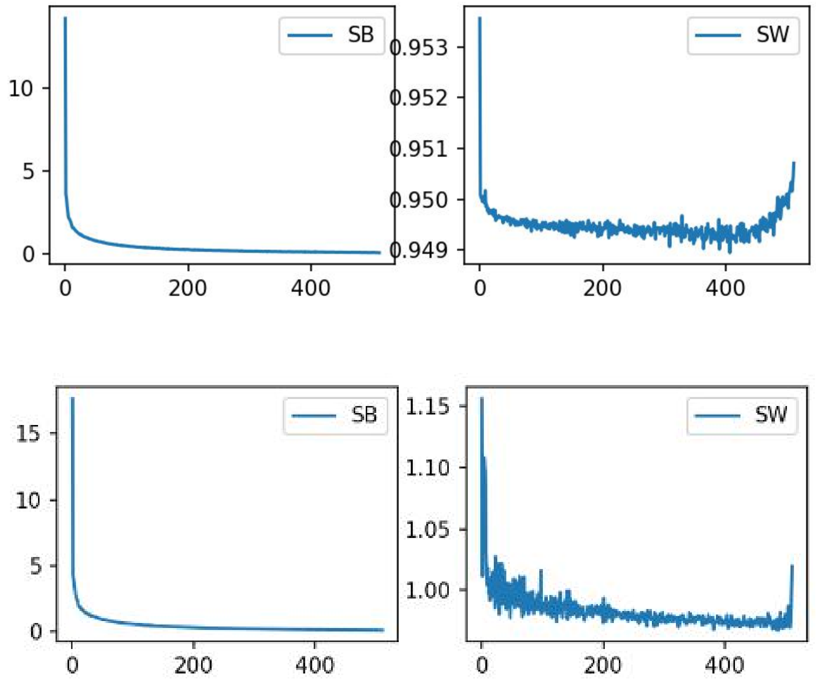
EER(%) with i-vector(400) on SITW

	Optimal results(EER)
Basic NL	8. 20%
+LN	<u>6. 32%</u>
+SD/LT	7. 68%

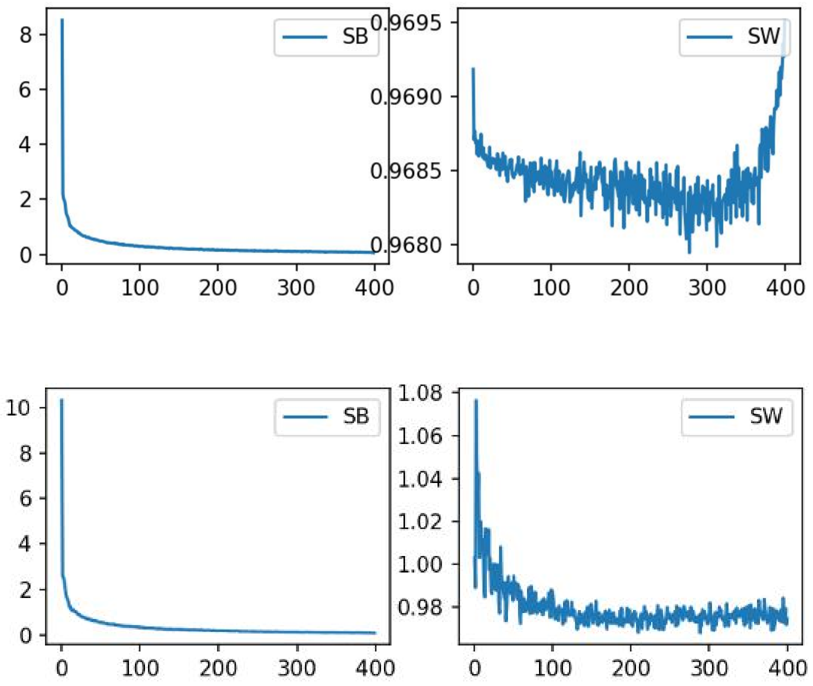
Statistics analysis

- Why length-norm doesn't perform better than SD/LT.

x-vector(512) on ***CNC***



i-vector(400) on ***CNC***

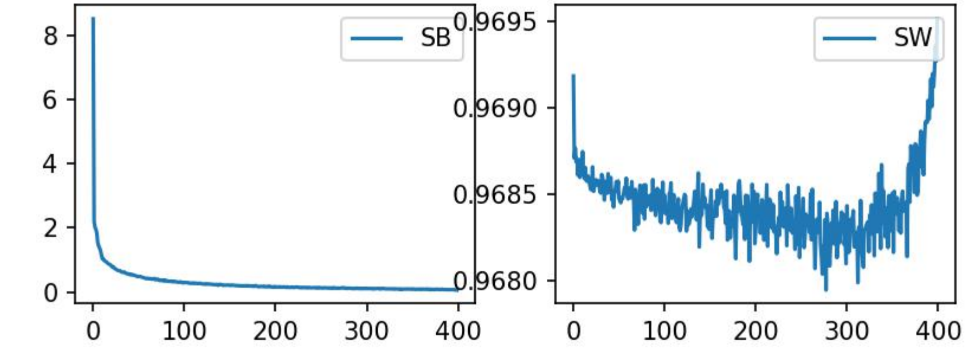


first line : SD/LT

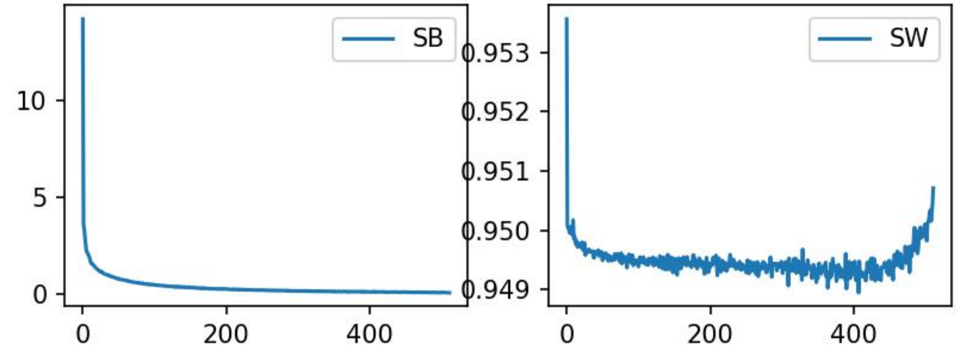
second line : length-norm

Is within-var really equal to 1?

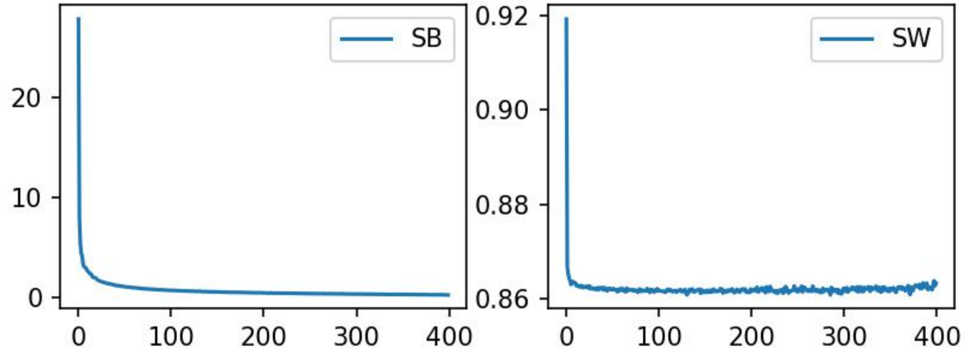
i-vector(400) on CNC



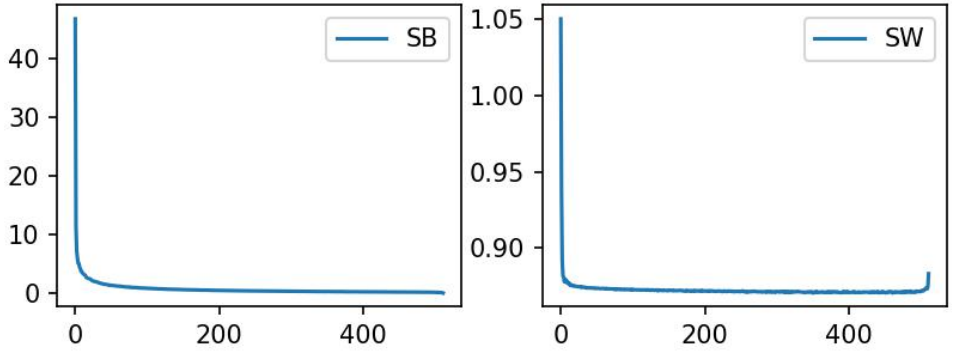
x-vector(512) on CNC



i-vector(400) on Voxceleb



x-vector(512) on Voxceleb



Theoretical vs. statistical

- Considering that within-var is assumed to be equal to 1 in the scoring, which is not rigorous, so we replace within-var in the scoring of different experiments.

EER(%) with ***x-vector***(512) on ***CNC.eval***

	Optimal(EER)		Optimal (EER)
Basic(NL)	19. 79%		
+LN	12. 71%	SD/LT	12. 63%
+*LN	12. 60%	***SD/LT**	**11. 95%**

EER(%) with ***x-vector***(512) on ***SITW***

	Optimal(EER)		Optimal (EER)
Basic(NL)	6. 86%		
+LN	4. 51%	SD/LT	5. 91%
+*LN	4. 43%	***SD/LT**	**4. 27%**

EER(%) with i-***vector***(400) on ***CNC.eval***

	Optimal(EER)		Optimal (EER)
Basic(NL)	15. 56%		
+LN	15. 69%	SD/LT	15. 44%
+*LN	15. 70%	***SD/LT**	**15. 44%**

EER(%) with i-***vector***(400) on ***SITW***

	Optimal(EER)		Optimal (EER)
Basic(NL)	8. 20%		
+LN	6. 32%	SD/LT	7. 68%
+*LN	6. 29%	***SD/LT**	**6. 12%**

Combine LN+SD/LT

- Combine LN+SD/LT and what will the performance be papered

EER(%) with ***x-vector***(400) on ***CNC.eval***

	Optimal(EER)
Basic(NL)	19. 79%
LN	12. 71%
LN +SD/LT	12. 71%

EER(%) with ***i-vector***(512) on ***CNC.eval***

	Optimal(EER)
Basic(NL)	15. 56%
LN	15. 69%
LN +SD/LT	15. 69%

EER(%) with ***x-vector***(512) on ***SITW***

	Optimal(EER)
Basic(NL)	6. 86%
LN	4. 51%
LN +SD/LT	4. 54%

EER(%) with ***i-vector***(400) on ***SITW***

	Optimal(EER)
Basic(NL)	8. 20%
LN	6. 32%
LN +SD/LT	6. 18%

Summary

- SD/LT works better than basic NL scoring.
- X-vector performs better than i-vector.
- Length-norm doesn't do very well on the complex data set.
- It is effective to replace theoretical statistics with actual statistics and has the best performance on SD/LT.
- Length-norm + SD/LT is useless.

Thank you !