Block-wise training for I-vector

Bie Fanhu
CSLT, RIIT, THU
2014-06-23

Outline

- Introduction
- Review of I-vector Theory
- Block-wise Training
- Experiments
- Conclusions

Introduction

- I-vector
- Time cost: $O(CDM + CM^2 + M^3)$

  where C is the number of Gaussian components, D is the dimension of features, M is the dimension of i-vector space.

Review of I-vector Theory

- I-vector space: a low space which is assumed to contain both the speaker and session
- Loading matrix: a transformation which associate the high dimension space and the I-vector space
- I-vector: a vector located in I-vector space which was extracted and represented for a speech of a speaker

I-vector algorithm

- $M = m + Tw$
- m is the supervector which was concatenating all the mean vector of each component of the UBM.
- T is the loading matrix
- w is the i-vector which normally distributed in the i-vector space
- Decomposed to each component, the formula can be written as: $M_c = m_c + T_c w$

I-vector extraction

- Given a speech segment X, i-vector is computed as the mean of the posterior probability p(w|X)
- The zero- and first-order statistics of the speech associated to each component c is defined as follows:

$$N_c = \sum_{t=1} P(c|\mathbf{x}_t)$$

$$\mathbf{F}_c = \sum_{t=1} P(c|\mathbf{x}_t)(\mathbf{x}_t - \mathbf{m}_c)$$

I-vector extraction

Since the prior p(w) is a Gaussian, the posterior of w is a Gaussian as follows:

$$p(\mathbf{w}|\mathbf{X}) \sim \mathcal{N}(\bar{\mathbf{w}}, \boldsymbol{\Xi})$$

$$\bar{\mathbf{w}} = (\mathbf{I} + \sum_c \mathbf{T}_c^T \boldsymbol{\Sigma}_c^{-1} \mathbf{N}_c \mathbf{T}_c)^{-1} \mathbf{T}^T \boldsymbol{\Sigma}^{-1} \mathbf{F}$$

$$\boldsymbol{\Xi} = (\mathbf{I} + \sum_c \mathbf{T}_c^T \boldsymbol{\Sigma}_c^{-1} \mathbf{N}_c \mathbf{T}_c)^{-1}$$

where $\mathbf{N}_c = N_c \mathbf{I}$.

I-vector extraction

- The loading matrix can be trained by an EM procedure
- The E step computes posterior probability p(w|X)
- The M step optimizes T by maximizing the following likelihood function:

$$\sum_u \sum_c \sum_t (\mathbf{x}_t - \mathbf{T}_c\mathbf{w} - \mathbf{m}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{x}_t - \mathbf{T}_c\mathbf{w} - \mathbf{m}_c).$$

- Update T with the following formula:

$$\mathbf{T}_c = (\sum_u \mathbf{F}_c(u)\bar{\mathbf{w}})(\sum_u \mathbf{N}_c(u)(\bar{\mathbf{w}}\bar{\mathbf{w}}^T + \boldsymbol{\Xi})).$$

Motivation for Block-wise Training

- The mean of different component is uncorrelated or little correlated
- The different dimension of the mean vector is uncorrelated or little correlated
- The EM procedure for the computation of the i-vector and the loading matrix is largely block-wise

Block-wise Training

Two factors which takes great cost of time:

- Utterance specific zero-order statistic N_c
- Large dimension of loading matrix T

Reducing dimension of Loading matrix T

- Component block-wise training
- Dimension block-wise training

Block-wise Training

Component block-wise training

$$
[\, M_1\ M_2\ \cdots\ M_{c/2-1}\ \ M_{c/2}\ M_{c/2+1}\ \cdots\ M_c \,]^{*} = \begin{pmatrix} T_1 & 0 \\ T_2 & 0 \\ \cdots & 0 \\ T_{c/2-1} & 0 \\ 0 & T_{c/2} \\ 0 & T_{c/2+1} \\ 0 & \cdots \\ 0 & T_c \end{pmatrix} .
$$

Block-wise Training

Dimension block-wise training

Assume that each mean is $[M_{c,D1}, M_{c,D2}]$:

$$[M_{1,D1}\ M_{1,D2}\ M_{1,D1}\ M_{1,D2}\ \cdots\ M_{1,D1}\ M_{1,D2}]^{*} = \begin{pmatrix} T_{1,D1} & 0 \\ 0 & T_{1,D2} \\ T_{2,D1} & 0 \\ 0 & T_{2,D2} \\ \cdots & \cdots \\ T_{c,D1} & 0 \\ 0 & T_{c,D2} \end{pmatrix}.$$

Loading Matrix Divided

Two factors which takes great cost of time:

- Utterance specific zero-order statistic N_c
- Large dimension of loading matrix T

Reducing dimension of Loading matrix T

- Component block-wise training
- Dimension block-wise training

Block-wise Training

- We assume the different blocks are uncorrelated
- The posterior and loading matrix can be estimated with a divide-and-combine way:
  - Divide: Estimate the data according to the groups
  - Combine: Then the data were combined to form the posterior and loading matrix

Speeding up of Block-wise Training

- Reduce the dimension of the i-vector
- Reduce the numbers of the component, which reduces the dimension of the loading matrix
- Because the blocks are uncorrelated, the training process can be parrelel

Experiment Result

Database:

- UBM, loading matrix: Female data in NIST SRE05, about 160 hours
- Train and verify: NIST SRE06 core test, 460 speakers, 2127 test segments, totally 29153 trials.

Experiment

Database:

- UBM, loading matrix: Female data in NIST SRE05, about 160 hours
- Train and verify: NIST SRE06 core test, 460 speakers, 2127 test segments, totally 29153 trials.

Configuration:

- Features: 20 + 20 + 10 of MFCC including 1 dimension of energy
- UBM: 1024 Gaussian Components
- I-vector: 500 or 1000 dimensions

Component-wise Training

	Training time (hours)	i-vec dim	EER%	
			dir.	comp.
Baseline1	14.1	500	9.79	-
Baseline2	60.2	1000	14.99	-
GA1	3.53	250	11.13	-
GA2	3.50	250	11.17	-
GA1 + GA2	-	500	10.38	10.68
GB1	7.80	500	10.38	-
GB2	7.80	500	10.07	-
GB1 + GB2	-	1000	**9.59**	10.03

DimensionComponent-wise Training

	Training time (hours)	i-vec dim	EER% dir.	EER% comp.
Baseline1	14.1	500	9.79	-
Baseline2	60.2	1000	14.99	-
GA1	6.0	250	10.53	-
GA2	6.0	250	12.16	-
GA1 + GA2	-	500	10.39	10.23
GB1	13.5	500	9.69	-
GB2	13.5	500	11.57	-
GB1 + GB2	-	1000	10.78	**9.29**

Conclusions

- We divided the loading matrix into blocks and train the block matrices independent and parallel way
- While the block structure can trade off the training data and model complexity approaches, it will improve the performance a little.

Conclusions

Thanks for your attention!