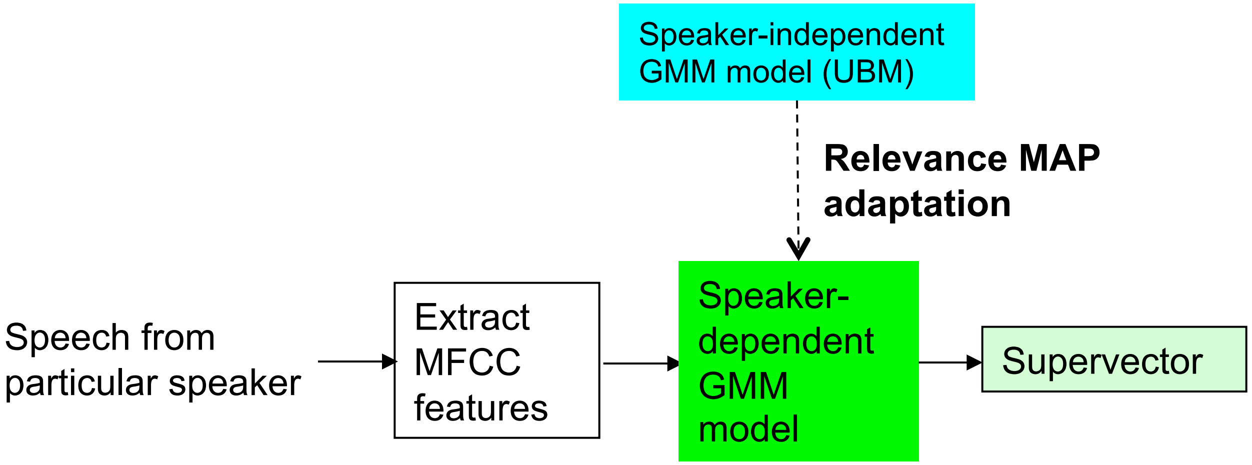
Joint Factor Analysis (JFA) and i-vector Tutorial

Howard Lei

Supervectors for speakers


Speaker-independent GMM model (UBM)
Relevance MAP adaptation
Speech from particular speaker
Extract MFCC features
Speaker-dependent GMM model
Supervector

- Relevance MAP adaptation is a linear interpolation of all mixture components of UBM to increase likelihood of speech from particular speaker

- Supervectors consist of the speaker-dependent GMM mean components

- **Problem:** Relevance MAP adaptation adapts to not only speaker-specific characters of speech, but also channel and other nuisance factors.

- Hence, supervectors generated in this way are non-ideal.

JFA Intuition

- A supervector for a speaker should be decomposable into speaker independent, speaker dependent, channel dependent, and residual components

- Each component can be represented by a low-dimensional set of factors, which operate along the principal dimensions (i.e. eigen-dimensions) of the corresponding component

- For instance, the following illustrates the speaker dependent component (known as the eigenvoice component) and corresponding factors:

$$V * y = \begin{bmatrix} | & | & | & | \\ v_1 & v_2 & \cdots & v_N \\ | & | & | & | \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Eigenvoice matrix

Low-dimensional eigenvoice (or speaker) factors

Each speaker factor controls an eigen-dimension of the eigenvoice matrix (i.e. y_1 controls v_1)

JFA Model

- A given speaker GMM supervector s can be decomposed as follows:

$$s = m + Vy + Ux + Dz$$

"Ideal" speaker supervector

Speaker-independent component

Speaker-dependent component

Channel-dependent component

Speaker-dependent residual component

- where:
  - Vector m is a speaker-independent supervector (from UBM)
  - Matrix V is the eigenvoice matrix
  - Vector y is the speaker factors. Assumed to have N(0,1) prior distribution
  - Matrix U is the eigenchannel matrix
  - Vector x is the channel factors. Assumed to have N(0,1) prior distribution
  - Matrix D is the residual matrix, and is diagonal
  - Vector z is the speaker-specific residual factors. Assumed to have N(0,1) prior distribution

Dimensions of JFA model

- For a 512-mixture GMM-UBM system, the dimensions of each JFA component are typically as follows:
  - Matrix V: 20,000 by 300 (300 eigenvoice components)
  - Vector y: 300 by 1 (300 speaker factors)
  - Matrix U: 20,000 by 100 (100 eigenchannel components)
  - Vector x: 100 by 1 (100 channel factors)
  - Matrix D: 20,000 by 20,000 (20,000 residual components)
  - Vector z: 20,000 by 1 (20,000 speaker-specific residual components)
- These dimensions have been empirically determined to produce best results

Training the JFA model

- Note: the following is taken from the paper “A Study of Inter-Speaker Variability in Speaker Verification” by Kenny et. al, 2008.

- We train the JFA matricies in the following order:
  1. Train the eigenvoice matrix V, assuming that U and D are zero
  2. Train the eigenchannel matrix U given estimate of V, assuming that D is zero
  3. Train residual matrix D given estimates of V and U

- Using these matrices, we compute the y (speaker), x (channel), and z (residual) factors

- We compute the final score using the matricies and factors

Training the V matrix

1) Accumulate 0th, 1st, and 2nd order sufficient statistics for each speaker (s) and Gaussian mixture component (c)

The posterior of Gaussian component c for observation t of speaker s

0th order statistic

$$N_c(s) = \sum_{t \in s} \gamma_t(c)$$

Sum over feature frames of all relevant conversation sides of speaker s

1st order statistic

$$F_c(s) = \sum_{t \in s} \gamma_t(c) Y_t$$

2nd order statistic

$$S_c(s) = diag\left(\sum_{t \in s} \gamma_t(c) Y_t Y_t^* \right)$$

Denotes Hermitian transpose of vector or matrix

Keep only the diagonal entries; zero out other entries

Training the V matrix

2) Center the 1st and 2nd order statistics

0th order statistic

UBM mean for mixture component c

Centered 1st order statistic

$$\tilde{F}_c(s) = F_c(s) - N_c(s) m_c$$

$$\tilde{S}_c(s) = S_c(s) - diag\left(F_c(s) m_c^* + m_c F_c(s)^* - N_c(s) m_c m_c^*\right)$$

Centered 2nd order statistic

Training the V matrix

3) Expand the statistics into matricies

$$NN(s) = \begin{bmatrix} N_1(s) * I & & \\ & \ddots & \\ & & N_C(s) * I \end{bmatrix}$$

$$FF(s) = \begin{bmatrix} \tilde{F}_1(s) \\ \vdots \\ \tilde{F}_C(s) \end{bmatrix}$$

$$SS(s) = \begin{bmatrix} \tilde{S}_1(s) & & \\ & \ddots & \\ & & \tilde{S}_C(s) \end{bmatrix}$$

Identity matrix

C total Gaussian mixtures

Training the V matrix

4) Initial estimate of the speaker factors y

Inverse UBM covariance matrix

Use random initialization of V

$$l_V(s) = I + V^* * \Sigma^{-1} * NN(s) * V$$

Posterior distribution of y(s) given all data from speaker s

$$y(s) \sim Normal(l_V^{-1}(s) * V^* * \Sigma^{-1} * FF(s), l_V^{-1}(s)) \Rightarrow$$

Gaussian normal distribution

$$\bar{y}(s) = E[y(s)] = l_V^{-1}(s) * V^* * \Sigma^{-1} * FF(s)$$

The expected value (the value we want)

Like a least-squares estimate to:

$$\min_{y(s)} || FF(s) - Vy(s) ||^2$$

Training the V matrix

5) Accumulate some additional statistics across the speakers

$$N_c = \sum_s N_c(s)$$

Covariance of posterior distribution of y(s)

$$\mathrm{A}_c = \sum_s N_c(s) l_V^{-1}(s)$$

$$\mathbb{C} = \sum_s FF(s) * (l_V^{-1}(s) * V^* * \Sigma^{-1} * FF(s))^*$$

Transposed mean of posterior distribution of y(s)

$$NN = \sum_s NN(s)$$

Training the V matrix

6) Compute V estimate

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_C \end{bmatrix} = \begin{bmatrix} A_1^{-1} * \mathbb{C}_1 \\ \vdots \\ A_C^{-1} * \mathbb{C}_C \end{bmatrix} \quad \text{where} \quad \mathbb{C} = \begin{bmatrix} \mathbb{C}_1 \\ \vdots \\ \mathbb{C}_C \end{bmatrix}$$

Block matrix components of V corresponding to each Gaussian mixture

Block matrix components of C corresponding to each Gaussian mixture

Training the V matrix

7) Compute covariance update (optional)

$$\Sigma = NN^{-1}\left(\left(\sum_{s} SS(s)\right) - diag(\mathbb{C} * V^{*})\right)$$

8) Run approx. 20 iterations of steps 4-6 (or 4-7). Substitute estimate of V into equations in step 4.

Training the U matrix

1) Compute estimate of speaker factor y for each speaker, and 0th and 1st order statistics for each conversation side (conv) of each speaker (s) in JFA training data

$$N_c(conv,s) = \sum_{t \in conv,s} \gamma_t(c)$$

$$F_c(conv,s) = \sum_{t \in conv,s} \gamma_t(c) Y_t$$

Training the U matrix

2) For each speaker (s), compute the speaker shift (along with speaker-independent shift) using matrix V and speaker factors y

$$spkrshift(s) = m + V * y(s)$$

3) For each conversation side of each speaker (used for JFA training), subtract Gaussian posterior-weighted speaker shift from first order statistics

$$\tilde{F}_c(conv, s) = F_c(conv, s) - spkrshift(s) * N_c(conv, s)$$

Training the U matrix

4) Expand the statistics into matricies

$$NN(conv,s) = \begin{bmatrix} N_1(conv,s) * I & & \\ & \ddots & \\ & & N_C(conv,s) * I \end{bmatrix}$$

$$FF(conv,s) = \begin{bmatrix} \tilde{F}_1(conv,s) \\ \vdots \\ \tilde{F}_C(conv,s) \end{bmatrix}$$

Training the U matrix

5) NN(conv,s) and FF(conv,s) used to train U and x in exact same way that NN(s) and FF(s) was used to train V and y

6) Run approx. 20 iterations of training procedure for V and y using NN(conv,s) and FF(conv,s)

Intuition: For the V matrix, we focused on obtaining the speaker-based principal dimensions. For the U matrix, we focus on obtaining the channel (or non-speaker, or nuisance)-based principal dimensions. Hence, we use the speaker-subtracted statistics to train U in the same way the speaker statistics were used to train V

Training the D matrix

1) For each speaker (s), compute the speaker shift using matrix V and speaker factors y

$$spkrshift(s) = m + V * y(s)$$

2) For each conversation side (conv) of speaker (s), compute the channel shift using matrix U and channel factors z

$$chanshift(conv, s) = U * x(conv, s)$$

3) For each speaker (used for JFA training), subtract Gaussian posterior-weighted speaker shift AND channel shifts from first order statistics

$$\tilde{F}_c(s) = F_c(s) - spkrshift(s) * N_c(s) - \sum_{conv \in s} \underset{\uparrow}{chanshift}(conv, s) * N_c(conv, s)$$

Computed for V estimate

Training the D matrix

4) Expand the statistics into matricies

$$NN(s) = \begin{bmatrix} N_1(s) * I & & \\ & \ddots & \\ & & N_C(s) * I \end{bmatrix}$$

$$FF(s) = \begin{bmatrix} \tilde{F}_1(s) \\ \vdots \\ \tilde{F}_C(s) \end{bmatrix}$$

Training the D matrix

5) Initial estimate of the residual factors z

Use random initialization of D

$$l_D(s) = I + D^2 * \Sigma^{-1} * NN(s)$$

$$z(s) \sim Normal(l_D^{-1}(s) * D * \Sigma^{-1} * FF(s), l_D^{-1}(s)) \Rightarrow$$

$$\overline{z}(s) = E[z(s)] = l_D^{-1}(s) * D * \Sigma^{-1} * FF(s)$$

The expected value (the value we want)

Training the D matrix

6) Accumulate some additional statistics across the speakers

$$N_c = \sum_s N_c(s)$$

Covariance of posterior distribution of y(s)

$$a = \sum_s diag(NN(s) * l_D^{-1}(s))$$

$$b = \sum_s diag(FF(s) * (l_D^{-1}(s) * D * \Sigma^{-1} * FF(s))^{*})$$

Transposed mean of posterior distribution of y(s)

$$NN = \sum_s NN(s)$$

Training the D matrix

7) Compute D estimate

$$D = \begin{bmatrix} D_1 \\ \vdots \\ D_C \end{bmatrix} = \begin{bmatrix} a_1^{-1} * b_1 \\ \vdots \\ a_C^{-1} * b_C \end{bmatrix} \quad \text{where} \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_C \end{bmatrix}$$

Block matrix components of D corresponding to each Gaussian mixture

Block matrix components of b corresponding to each Gaussian mixture

8) Iterate steps 5-7 20 times. Substitute estimate of D into equations in step 5.

Computing linear score

- Refer to “Comparison of Scoring Methods used in Speaker Recognition with Joint Factor Analysis” by Glembek, et. al.

1) Use the matricies V, U, and D to get estimates of y, x, and z, in terms of their posterior means given the observations

2) For test conversation side (tst) and target speaker conversation side (tar), one way to obtain final score is via the following linear product:

$$Score = (V * y(tar) + D * z(tar))^{*} * \Sigma^{-1} * (FF(tst) - NN(tst) * m - NN(tst) * U * x(tst))$$

Target speaker conversation side centered around speaker and residual factors

Test conversation side has speaker-independent and channel factors removed, and hence also centered around speaker and residual factors

The i-vector approach

An i-vector system uses a set of low-dimensional total variability factors (w) to represent each conversation side. Each factor controls an eigen-dimension of the total variability matrix (T), and are known as the i-vectors.

$$\underset{\substack{\text{Conversation}\\ \text{side supervector}}}{s} = m + \underset{\substack{\text{Total-variability}\\ \text{matrix}}}{T}\,\underset{\text{i-vector}}{w}$$

1) To train T, run exact training procedure used to train V, but treat all conversation sides of all training speakers as belonging to different speakers

2) Given T, obtain i-vectors (w) for each conversation side

The i-vector approach

3) For channel compensation of i-vectors, perform LDA then WCCN (techniques empirically determined to perform well) on i-vectors. Denote channel-compensated i-vectors as ω.

4) Perform cosine distance scoring (CDS) on channel-compensated i-vectors ω for a pair of conversation sides:

$$score(\omega_1, \omega_2) = \frac{\omega_1^{*} * \omega_2}{\| \omega_1 \| * \| \omega_2 \|} \qquad = \cos(\theta_{\omega_1, \omega_2})$$

If i-vectors of two speakers point in the same direction, their cosine distance takes highest possible value of 1. If they point in opposite directions, their cosine distance takes lowest possible value of -1.

The End