Useful Derivations for i-Vector Based Approach to Data Clustering in Speech Recognition

Yu Zhang

December 16, 2011

Contents

1 Background

1.1 Traditional maximum likelihood eigen-decomposition

Let $\boldsymbol{M}(s)$ denote the mean supervector for a speaker s formulating as:

$$\boldsymbol{M}(s) = \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}, \tag{1}$$

and the likelihood function written as:

$$\prod_s \max_{\boldsymbol{w}} P_{\text{HMM}}(\boldsymbol{Y}_s | \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}, \boldsymbol{R}). \tag{2}$$

The optimization proceeds by iterating the following two steps:

1. For each training speaker s, use the current estimates of $\boldsymbol{T}$ and $\boldsymbol{R}$ to find the $\boldsymbol{w}$ which maximizes the HMM likelihood given the speaker's training data $\boldsymbol{Y}_s$:
$$\boldsymbol{w}(s) = \arg\max_{\boldsymbol{w}} P_{\text{HMM}}(\boldsymbol{Y}_s | \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}, \boldsymbol{R}) \tag{3}$$
2. Update $\boldsymbol{T}$ and $\boldsymbol{R}$ by maximizing:
$$\prod_s P_{\text{HMM}}(\boldsymbol{Y}_s | \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}(s), \boldsymbol{R}) \tag{4}$$

1.2 What's new in Kenny's paper?

In Kenny's paper [3], they treat $\boldsymbol{w}(i)$ as a random vector with a standard normal distribution. So the estimation of $\boldsymbol{T}$ and $\boldsymbol{R}$ becomes to maximize the marginal likelihood function

$$\prod_s \int_{\boldsymbol{w}} P(\boldsymbol{Y}_s | \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}, \boldsymbol{R}) \mathcal{N}(\boldsymbol{w} | \boldsymbol{0}, \boldsymbol{I}) \mathrm{d}\boldsymbol{w}, \tag{5}$$

where the product extends over all speakers in the training set.

Calculating the posterior distribution of $\boldsymbol{w}(i)$ in the E-step rather than the maximum likelihood estimate is the key to avoiding the degeneracy problem that arises when Eq. (2) is used to estimate the eigenvoices in situations where speaker-dependent training is not feasible (e.g., when training data is sparse) or the number of eigenvoices is large compared with the number of training speakers.

The algorithm can be briefly break-down into the following steps:

1. For each training speaker s, use the current alignment of the speaker's training data and the current estimates $\boldsymbol{T}$ and $\boldsymbol{R}$ to carry out MAP speaker adaptation. Use the speaker adapted model to realign the speaker's training data.
2. The **E-step**: For each speaker s, calculate the posterior distribution of $\boldsymbol{w}(i)$ using the current alignment of the speaker's training data, the current estimates of $\boldsymbol{T}$ and $\boldsymbol{R}$ and the prior $\mathcal{N}(\boldsymbol{w} | \boldsymbol{0}, \boldsymbol{I})$.
3. The **M-step**: Update $\boldsymbol{T}$ and $\boldsymbol{R}$ by a linear regression in which the $\boldsymbol{w}(i)$'s play the role of the explanatory variables.

2 The i-vector based approach to data clustering

We are considering in this memo to use i-vector based approach for more generic data clustering problems in speech recognition. Possible applications include training data clustering for multiple acoustic model training [7] and acoustic sniffing in IVN-based training [5]. So hereafter, we will change the subscript s representing the speaker to subscript i representing each individual acoustic unit that to be clustered. The granularity of the acoustic units can be one single utterance, several utterances (e.g., one conversational side in Switchboard), or even more utterances from a pre-defined source (speaker, environment, channel, etc.)

2.1 Data model

Suppose we are given a set of training data denoted as $\mathcal{Y} = \{\boldsymbol{Y}^i | i = 1, 2, \ldots, I\}$, where $\boldsymbol{Y}^i = (\boldsymbol{y}_1^i, \boldsymbol{y}_2^i, \ldots, \boldsymbol{y}_{T_i}^i)$ is a sequence of D-dimensional feature vectors extracted from the i-th acoustic unit (e.g., utterance). From $\mathcal{Y}$, a Gaussian Mixture Model (GMM) can be trained using a Maximum Likelihood (ML) criterion to serve as a Universal Background Model (UBM). Let's use $\boldsymbol{\Omega} = \{c_k, \boldsymbol{m}_k, \boldsymbol{R}_k | k = 1, \ldots, K\}$ to denote the set of UBM-GMM parameters where c_k's are mixture component weights, $\boldsymbol{m}_k$ and $\boldsymbol{R}_k$ are D-dimensional mean and $D \times D$ diagonal covariance matrix for the k^{th} mixture component.

Given the data model $\boldsymbol{\Omega}$, the probability of each acoustic unit $\boldsymbol{Y}^i$ can be written as:

$$p(\boldsymbol{Y}^i | \boldsymbol{\Omega}) = \prod_{t=1}^{T_i} \sum_{k=1}^{K} c_k \mathcal{N}(\boldsymbol{y}_t^i; \boldsymbol{m}_k, \boldsymbol{R}_k). \tag{6}$$

And we denote $\boldsymbol{\Omega}^{(0)}$ hereafter the initial model parameters of the UBM.

2.2 Frontend i-vector extraction

2.2.1 The i-vector solution

Let $\boldsymbol{M}_0$ denote the $(D \cdot K)$-dimensional supervector by concatenating the $\boldsymbol{m}_k$'s. Given an utterance $\boldsymbol{Y}^i$, let's use another $(D \cdot K)$-dimensional random supervector $\boldsymbol{M}(i)$ to characterize it independent of its linguistic content. In i-vector based approach, $\boldsymbol{M}(i)$ is correlated with $\boldsymbol{M}_0$ as:

$$\boldsymbol{M}(i) = \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}(i), \tag{7}$$

where $\boldsymbol{T}$ is a fixed but unknown $(D \cdot K) \times F$ rectangular matrix of low rank (i.e., $F \ll (D \cdot K)$), and $\boldsymbol{w}(i)$ is an F-dimensional random vector having a prior distribution of $\mathcal{N}(\cdot; \boldsymbol{0}, \boldsymbol{I})$. A graphical model representation is shown in Fig. 1. In [2], $\boldsymbol{T}$ is called the *total variability matrix* and $\boldsymbol{w}(i)$ the *i-vector*. They are also comparable with the *loading matrix* and *factors* in factor analysis.

Figure 1: *A graphical model representation of i-vector approach.*

Given $\boldsymbol{Y}^i$, $\boldsymbol{M}(i)$, and $\boldsymbol{R}_k$'s, the i-vector is the MAP solution of the following problem:

$$\begin{aligned} \hat{\boldsymbol{w}}(i) &= \underset{\boldsymbol{w}(i)}{\operatorname{argmax}}\, p(\boldsymbol{w}(i)|\boldsymbol{Y}^i) \\ &= \underset{\boldsymbol{w}(i)}{\operatorname{argmax}}\, p(\boldsymbol{Y}^i|\boldsymbol{w}(i))p(\boldsymbol{w}(i)), \end{aligned} \tag{8}$$

where

$$p(\boldsymbol{Y}^i|\boldsymbol{w}(i)) = \prod_{t=1}^{T_i}\sum_{k=1}^{K} c_k \mathcal{N}(\boldsymbol{y}_t^i; \boldsymbol{M}_k(i), \boldsymbol{R}_k), \tag{9}$$

in which $\boldsymbol{M}_k(i)$ is the k-th D-dimensional subvector of $\boldsymbol{M}(i)$.

2.2.2 Smoothed data model

Eq. (9) is intractable because of the summation term. So Viterbi approximation was imposed in [3] so that

$$p(\boldsymbol{Y}^i|\boldsymbol{w}(i)) \simeq \prod_{t=1}^{T_i} \max_k \mathcal{N}(\boldsymbol{y}_t^i; \boldsymbol{M}_k(i), \boldsymbol{R}_k). \tag{10}$$

One can also define a "smoothed" version of $p(\boldsymbol{Y}^i|\boldsymbol{w}(i))$:

$$p(\boldsymbol{Y}^i|\boldsymbol{w}(i)) \simeq \prod_{t=1}^{T_i}\prod_{k=1}^{K} \mathcal{N}(\boldsymbol{y}_t^i; \boldsymbol{M}_k(i), \boldsymbol{R}_k)^{p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)})}, \tag{11}$$

where

$$p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)}) = \frac{c_k \mathcal{N}(\boldsymbol{y}_t^i; \boldsymbol{m}_k, \boldsymbol{R}_k^{(0)})}{\sum_{l=1}^{K} c_l \mathcal{N}(\boldsymbol{y}_t^i; \boldsymbol{m}_l, \boldsymbol{R}_l^{(0)})} \tag{12}$$

is the Baum-Welch occupancy probability of the k^{th} component given $\boldsymbol{y}_t^i$. Note that the production terms in Eq. (11) are not strictly probability densities because they do not integrate to 1. Using Eq. (11) will lead to the same i-vector solution as in [2], although in that paper, the form of $p(\boldsymbol{Y}^i|\boldsymbol{w}(i))$ was never explicitly given.

2.2.3 Solving i-vector

Given the smoothed data model in Eq. (11), the closed-form solution of the problem in Eq. (8) gives the i-vector extraction formula as follows:

$$\hat{\boldsymbol{w}}(i) = \boldsymbol{l}^{-1}(i)\boldsymbol{T}^\top \boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i), \tag{13}$$

where

$$\boldsymbol{l}(i) = \boldsymbol{I} + \boldsymbol{T}^\top \boldsymbol{\Gamma}(i)\boldsymbol{R}^{-1}\boldsymbol{T}, \tag{14}$$

$\boldsymbol{R}$ is the $(D \cdot K) \times (D \cdot K)$ block-diagonal matrix with $\boldsymbol{R}_k$ as its k^{th} block component, $\boldsymbol{\Gamma}(i)$ is a $(D \cdot K) \times (D \cdot K)$ block-diagonal matrix with $\gamma_k(i)\boldsymbol{I}_{D\times D}$ as its k^{th} block component; $\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)$ is a $(D \cdot K)$-dimensional supervector with $\boldsymbol{\Gamma}_{\boldsymbol{y},k}(i)$ as its k^{th} D-dimensional subvector. The statistics $\gamma_k(i)$ and $\boldsymbol{\Gamma}_{\boldsymbol{y},k}(i)$ are calculated as follows:

$$\begin{aligned} \gamma_k(i) &= \sum_{t=1}^{T_i} p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)}), \\ \boldsymbol{\Gamma}_{\boldsymbol{y},k}(i) &= \sum_{t=1}^{T_i} p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)})(\boldsymbol{y}_t^i - \boldsymbol{m}_k). \end{aligned} \tag{15}$$

Proof: First,

$$\begin{aligned} &\log p(\boldsymbol{Y}^i|\boldsymbol{w}(i)) = \sum_{t=1}^{T_i}\sum_{k=1}^{K} p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)}) \log \mathcal{N}(\boldsymbol{y}_t^i; \boldsymbol{M}_k(i), \boldsymbol{R}_k) \\ &= \sum_{t=1}^{T_i}\sum_{k=1}^{K} p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)}) \big[\log \frac{1}{(2\pi)^{D/2}|\boldsymbol{R}_k|^{1/2}} \\ &\quad -\frac{1}{2}(\boldsymbol{y}_t^i - \boldsymbol{m}_k - \boldsymbol{T}_k\boldsymbol{w}(i))^\top \boldsymbol{R}_k^{-1}(\boldsymbol{y}_t^i - \boldsymbol{m}_k - \boldsymbol{T}_k\boldsymbol{w}(i))\big] \\ &= \sum_{t=1}^{T_i}\sum_{k=1}^{K} p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)}) \big[\log \frac{1}{(2\pi)^{D/2}|\boldsymbol{R}_k|^{1/2}} - \frac{1}{2}(\boldsymbol{y}_t^i - \boldsymbol{m}_k)^\top \boldsymbol{R}_k^{-1}(\boldsymbol{y}_t^i - \boldsymbol{m}_k) \\ &\quad +\boldsymbol{w}^\top(i)\boldsymbol{T}_k^\top \boldsymbol{R}_k^{-1}(\boldsymbol{y}_t^i - \boldsymbol{m}_k) - \frac{1}{2}\boldsymbol{w}^\top(i)\boldsymbol{T}_k^\top \boldsymbol{R}_k^{-1}\boldsymbol{T}_k\boldsymbol{w}(i)\big]. \end{aligned} \tag{16}$$

Only the last two terms are related with $\boldsymbol{w}(i)$, and we can define:

$$\begin{aligned} \mathcal{H}(i) &= \sum_{t=1}^{T_i}\sum_{k=1}^{K} p(k|\boldsymbol{y}_t^i, \boldsymbol{\Omega}^{(0)}) \big[\boldsymbol{w}^\top(i)\boldsymbol{T}_k^\top \boldsymbol{R}_k^{-1}(\boldsymbol{y}_t^i - \boldsymbol{m}_k) - \frac{1}{2}\boldsymbol{w}^\top(i)\boldsymbol{T}_k^\top \boldsymbol{R}_k^{-1}\boldsymbol{T}_k\boldsymbol{w}(i)\big] \\ &= \boldsymbol{w}^\top(i)\boldsymbol{T}\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) - \frac{1}{2}\boldsymbol{w}^\top(i)\boldsymbol{T}^\top\boldsymbol{\Gamma}(i)\boldsymbol{R}^{-1}\boldsymbol{T}\boldsymbol{w}(i). \end{aligned} \tag{17}$$

So

$$\begin{aligned} &p(\boldsymbol{w}(i)|\boldsymbol{Y}^i) \propto p(\boldsymbol{Y}^i|\boldsymbol{w}(i))p(\boldsymbol{w}(i)) \\ &\quad \propto \exp\Big(\boldsymbol{w}^\top(i)\boldsymbol{T}\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) - \frac{1}{2}\boldsymbol{w}^\top(i)\boldsymbol{T}^\top\boldsymbol{\Gamma}(i)\boldsymbol{R}^{-1}\boldsymbol{T}\boldsymbol{w}(i)\Big) \cdot \exp\Big(-\frac{1}{2}\boldsymbol{w}^\top(i)\boldsymbol{w}(i)\Big) \\ &\quad = \exp\Big(\boldsymbol{w}^\top(i)\boldsymbol{T}\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) - \frac{1}{2}\boldsymbol{w}^\top(i)[\boldsymbol{T}^\top\boldsymbol{\Gamma}(i)\boldsymbol{R}^{-1}\boldsymbol{T} + \boldsymbol{I}]\boldsymbol{w}(i)\Big) \\ &\quad = \exp\Big(\boldsymbol{w}^\top(i)\boldsymbol{T}\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) - \frac{1}{2}\boldsymbol{w}^\top(i)\boldsymbol{l}(i)\boldsymbol{w}(i)\Big) \\ &\quad \propto \exp\Big(-\frac{1}{2}\big(\boldsymbol{w}(i) - \boldsymbol{l}^{-1}(i)\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)\big)^\top \boldsymbol{l}(i)\big(\boldsymbol{w}(i) - \boldsymbol{l}^{-1}(i)\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)\big)\Big). \end{aligned} \tag{18}$$

Therefore, the "posterior" distribution of $\boldsymbol{w}(i)$ is Gaussian of mean $\boldsymbol{l}^{-1}(i)\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)$ and covariance $\boldsymbol{l}^{-1}(i)$.

■

2.3 Hyperparameter estimation

Given the training data $\mathcal{Y}$, the hyperparameters $\boldsymbol{T}$ and $\boldsymbol{R}$ can be estimated by maximizing the following log-likelihood function:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{T},\boldsymbol{R}) &= \log\prod_{i=1}^{I}\int p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))\mathrm{d}\boldsymbol{w}(i)\\
&= \sum_{i=1}^{I}\log\int p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))\mathrm{d}\boldsymbol{w}(i)\\
&= \sum_{i=1}^{I}\log\int \frac{p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))}{p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))}p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))\mathrm{d}\boldsymbol{w}(i)\\
&= \sum_{i=1}^{I}\log\int \frac{p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))}{p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))}p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{w}(i)|\boldsymbol{Y}^i)\mathrm{d}\boldsymbol{w}(i)+C_1,
\end{aligned} \tag{19}$$

where $\boldsymbol{T}^{(0)}$ and $\boldsymbol{R}^{(0)}$ are hyperparameters from last iteration, C_1 is a constant independent of $\boldsymbol{T}$ and $\boldsymbol{R}$ (we will use C_j's hereafter for those constants). By Jensen's inequality, we have:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{T},\boldsymbol{R}) &\geq \sum_{i=1}^{I}\int\log\left(\frac{p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))}{p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))}\right)p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{w}(i)|\boldsymbol{Y}^i)\mathrm{d}\boldsymbol{w}(i)\\
&= \sum_{i=1}^{I}\int\log p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i,\boldsymbol{w}(i))p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{w}(i)|\boldsymbol{Y}^i)\mathrm{d}\boldsymbol{w}(i)+C_2\\
&= \sum_{i=1}^{I}\int\log p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i|\boldsymbol{w}(i))p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{w}(i)|\boldsymbol{Y}^i)\mathrm{d}\boldsymbol{w}(i)+C_3.
\end{aligned} \tag{20}$$

So an auxiliary function $\mathcal{A}$ can be built as:

$$\begin{aligned}
\mathcal{A}(\boldsymbol{T},\boldsymbol{R}) &= \sum_{i=1}^{I}\int\log p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i|\boldsymbol{w}(i))p_{\boldsymbol{T}^{(0)},\boldsymbol{R}^{(0)}}(\boldsymbol{w}(i)|\boldsymbol{Y}^i)\mathrm{d}\boldsymbol{w}(i)\\
&= \sum_{i=1}^{I}E\big(\log p_{\boldsymbol{T},\boldsymbol{R}}(\boldsymbol{Y}^i|\boldsymbol{w}(i))\big)
\end{aligned} \tag{21}$$

and the log likelihood function is

$$\log\mathcal{F}(\boldsymbol{T},\boldsymbol{R}) = \sum_i\mathcal{G}(i)-\frac{1}{2}|\boldsymbol{l}(i)|+\frac{1}{2}E[\boldsymbol{w}(i)^\top]\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) \tag{22}$$

2.3.1 Updating $\boldsymbol{T}$

Referring to Eqs (16) and (17), for updating $\boldsymbol{T}$, only the $\mathcal{H}$ function is involved. So given the properties in Eq. (48), we have

$$\begin{aligned}\mathcal{A} &= \sum_i E[\boldsymbol{w}^\top(i)\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) - \frac{1}{2}\boldsymbol{w}^\top(i)\boldsymbol{T}^\top\boldsymbol{\Gamma}(i)\boldsymbol{R}^{-1}\boldsymbol{T}\boldsymbol{w}(i)] + C_4 \\ &= \sum_i E[\boldsymbol{w}^\top(i)]\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) - \frac{1}{2}\mathrm{tr}(\boldsymbol{T}^\top\boldsymbol{\Gamma}(i)\boldsymbol{R}^{-1}\boldsymbol{T}E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)]) + C_4 \\ &= \sum_i \mathrm{tr}\left(\boldsymbol{R}^{-1}\Big(\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)] - \frac{1}{2}\boldsymbol{\Gamma}(i)\boldsymbol{T}E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)]\Big)\boldsymbol{T}^\top\right) + C_4,\end{aligned} \tag{23}$$

and

$$\frac{\partial\mathcal{A}}{\partial\boldsymbol{T}} = \sum_i \boldsymbol{R}^{-1}\Big(\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)] - \boldsymbol{\Gamma}(i)\boldsymbol{T}E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)]\Big). \tag{24}$$

Setting Eq. (24) to 0, and taking use of the diagonal structure of $\boldsymbol{\Gamma}(i)$, $\boldsymbol{T}$ can be solved line-by-line according to:

$$\boldsymbol{T}^m\sum_i \boldsymbol{\Gamma}^m(i)E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)] = \sum_i \boldsymbol{\Gamma}^m_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)], \tag{25}$$

in which $\boldsymbol{T}^m$, $\boldsymbol{\Gamma}^m$ and $\boldsymbol{\Gamma}^m_{\boldsymbol{y}}$ are the m^{th} row of $\boldsymbol{T}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}_{\boldsymbol{y}}$, respectively.

2.3.2 Updating $\boldsymbol{R}$

Go back to Eq. (16) and set

$$\begin{aligned}\mathcal{G}(i) &= \sum_{t=1}^{T_i}\sum_{k=1}^{K} p(k|\boldsymbol{y}_t^i,\boldsymbol{\Omega}^{(0)})\big[\log\frac{1}{(2\pi)^{D/2}|\boldsymbol{R}_k|^{1/2}} - \frac{1}{2}(\boldsymbol{y}_t^i-\boldsymbol{m}_k)^\top\boldsymbol{R}_k^{-1}(\boldsymbol{y}_t^i-\boldsymbol{m}_k)\big] \\ &= \frac{1}{2}\sum_{k=1}^{K}\gamma_k(i)\log|\boldsymbol{R}_k^{-1}| - \mathrm{tr}(\boldsymbol{R}_k^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}\boldsymbol{y}^\top,k}(i)) + C_5,\end{aligned} \tag{26}$$

in which

$$\boldsymbol{\Gamma}_{\boldsymbol{y}\boldsymbol{y}^\top,k}(i) = \sum_{t=1}^{T_i} p(k|\boldsymbol{y}_t^i,\boldsymbol{\Omega}^{(0)})(\boldsymbol{y}_t^i-\boldsymbol{m}_k)(\boldsymbol{y}_t^i-\boldsymbol{m}_k)^\top. \tag{27}$$

It is not difficult to see that:

$$\mathcal{A} = \sum_i \mathcal{G}(i) + E[\mathcal{H}(i)]. \tag{28}$$

Because

$$\frac{\partial\mathcal{G}(i)}{\partial\boldsymbol{R}^{-1}} = \frac{1}{2}\sum_{k=1}^{K}\gamma_k(i)\boldsymbol{R} - \boldsymbol{\Gamma}_{\boldsymbol{y}\boldsymbol{y}^\top,k}(i), \tag{29}$$

and

$$\frac{\partial E[\mathcal{H}(i)]}{\partial \boldsymbol{R}^{-1}} = \frac{\partial \mathrm{tr}\left(\boldsymbol{R}^{-1}\left(\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)] - \frac{1}{2}\boldsymbol{\Gamma}(i)\boldsymbol{T}E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)]\right)\boldsymbol{T}^\top\right)}{\partial \boldsymbol{R}^{-1}} \quad (30)$$

Note that Eq. (25)

$$\begin{aligned}\sum_i \frac{\partial E[\mathcal{H}(i)]}{\partial \boldsymbol{R}^{-1}} &= \frac{\partial \mathrm{tr}\left(\sum_i \boldsymbol{R}^{-1}\left(\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)] - \frac{1}{2}\boldsymbol{\Gamma}(i)\boldsymbol{T}E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)]\right)\boldsymbol{T}^\top\right)}{\partial \boldsymbol{R}^{-1}} \\ &= \frac{1}{2}\sum_i \left(\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)]\boldsymbol{T}^\top + \boldsymbol{T}E[\boldsymbol{w}(i)]\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)^\top\right) \quad (31)\end{aligned}$$

Setting $\frac{\partial \mathcal{A}}{\partial \boldsymbol{R}^{-1}}$ to 0, we have

$$\boldsymbol{R}_k = \frac{1}{\sum_i \gamma_k(i)}\left(\sum_i \boldsymbol{\Gamma}_{\boldsymbol{y}\boldsymbol{y}^\top,k}(i) - M_c\right) \quad (32)$$

where M_k denotes the kth diagonal block of the matrix

$$\frac{1}{2}\sum_i \left(\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)]\boldsymbol{T}^\top + \boldsymbol{T}E[\boldsymbol{w}(i)]\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)^\top\right). \quad (33)$$

2.3.3 Summarizing the EM algorithm

- **E-Step**:

$$\begin{aligned}E[\boldsymbol{w}(i)] &= \boldsymbol{l}^{-1}(i)\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) \\ E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)] &= E[\boldsymbol{w}(i)]E[\boldsymbol{w}^\top(i)] + \boldsymbol{l}^{-1}(i)\end{aligned} \quad (34)$$

- **M-Step**:

$$\begin{aligned}\boldsymbol{T}^m\sum_i \boldsymbol{\Gamma}^m(i)E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)] &= \sum_i \boldsymbol{\Gamma}^m_{\boldsymbol{y}}(i)E[\boldsymbol{w}^\top(i)] \\ \boldsymbol{R}_k &= \frac{1}{\sum_i \gamma_k(i)}\left(\sum_i \boldsymbol{\Gamma}_{\boldsymbol{y}\boldsymbol{y}^\top,k}(i) - M_k\right)\end{aligned} \quad (35)$$

3 Implementation

3.1 Training $\boldsymbol{T}$ and $\boldsymbol{R}$

1. Train a GMM-UBM using all the training data.
2. Initialize $\boldsymbol{R}$ by borrowing $\boldsymbol{R}_k$'s from the UBM, and $\boldsymbol{T}$ randomly as:

$$\boldsymbol{T}^{m,f} \in [-\alpha\boldsymbol{R}^{m,m}, \alpha\boldsymbol{R}^{m,m}], \quad \forall m = 1,\ldots,DK; f = 1,\ldots,F, \quad (36)$$

where $\boldsymbol{T}^{m,f}$ is the m^{th} line and f^{th} column of $\boldsymbol{T}$, which is similar for $\boldsymbol{R}^{m,m}$; α is typically set to 0.1.

3. For each acoustic unit i, extract the Baum-Welch statistics $\gamma_k(i)$, $\mathbf{\Gamma}_{\boldsymbol{y},k}(i)$ and $\mathbf{\Gamma}_{\boldsymbol{y}\boldsymbol{y}^\top,k}(i)$ using the UBM as in Eqs. (15) and (27).

4. The E-Step: Calculate the posterior expectation $E[\boldsymbol{w}(i)]$ and $E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)]$ using the statistics and the current estimation of $\boldsymbol{T}$ and $\boldsymbol{R}$ with Eq. (34).

5. Repeat Steps 3 and 4 for all i's. Accumulate necessary statistics for solving Eq. (35).

6. The M-Step: Updating $\boldsymbol{T}$ and $\boldsymbol{R}$ using Eq. (35).

7. If training converges, stop; otherwise go back to Step 4.

3.2 Extracting $\hat{\boldsymbol{w}}(i)$

Extracting $\hat{\boldsymbol{w}}(i)$ is straightforward by using the above Steps 3 and 4 and setting:

$$\hat{\boldsymbol{w}}(i) = E[\boldsymbol{w}(i)]. \tag{37}$$

The procedure is the same for both training and testing acoustic units.

3.3 Computational complexity

In i-vector extraction, the most computational cost is the calculation of $\boldsymbol{l}(i)$ in Eq. (14). The right-hand-side of the equation can be implemented step-by-step as:

$$\boldsymbol{R}^{-1} \rightarrow \mathbf{\Gamma}(i)\boldsymbol{R}^{-1} \rightarrow \boldsymbol{T}^\top\mathbf{\Gamma}(i)\boldsymbol{R}^{-1} \rightarrow \boldsymbol{T}^\top\mathbf{\Gamma}(i)\boldsymbol{R}^{-1}\boldsymbol{T}, \tag{38}$$

so the complexity is $O(I \times DK \times F \times F)$. In our implementation, the property that $\mathbf{\Gamma}(i)$ is a block diagonal matrix can be used to make the computation more efficient:

1. Compute all the K block matrices of $\boldsymbol{T}^\top\boldsymbol{R}^{-1}\boldsymbol{T}$:

$$\boldsymbol{H}_k = \boldsymbol{T}^\top\boldsymbol{R}_k\boldsymbol{T} \quad \forall k = 1, \ldots, K. \tag{39}$$

2. For each acoustic unit i,

$$\boldsymbol{l}(i) = \sum_k \gamma_k(i)\boldsymbol{H}_k. \tag{40}$$

By doing so, the complexity of evaluating Eq. (14) is dominated by $O(I \times K \times F \times F + DK \times F \times F)$.

4 Clustering of i-vectors using LBG algorithm

As described above, given the training corpus, an i-vector can be extracted from each acoustic unit. Given the training set i-vectors, we use a hierarchical divisive clustering algorithm, namely LBG algorithm [4], to cluster them into multiple clusters. To measure the similarity between two i-vectors $\hat{\boldsymbol{w}}(i)$ and $\hat{\boldsymbol{w}}(j)$, the both Euclidean distance and cosine similarity measure can be used.

It is quite easy to use Euclidean distance in clustering. However when cosine similarity is used, the calculation of the clustering centroid needs to be reviewed. Given the cosine similarity defined as:

$$sim(\hat{\boldsymbol{w}}(i), \hat{\boldsymbol{w}}(j)) = \frac{\hat{\boldsymbol{w}}^{\top}(i)\hat{\boldsymbol{w}}(j)}{||\hat{\boldsymbol{w}}(i)||||\hat{\boldsymbol{w}}(j)||}, \tag{41}$$

it can be proved that the corresponding clustering centroid, $\boldsymbol{c}^{\hat{\boldsymbol{w}}}$ consisting of n i-vectors $\hat{\boldsymbol{w}}(1), \hat{\boldsymbol{w}}(2), \ldots, \hat{\boldsymbol{w}}(n)$, can be calculated as:

$$\begin{aligned} \boldsymbol{c}^{\hat{\boldsymbol{w}}} &= \underset{\boldsymbol{c}}{\operatorname{argmax}} \sum_{i=1}^{n} sim(\hat{\boldsymbol{w}}(i), \boldsymbol{c}) \\ &= \begin{cases} \dfrac{\sum_{i=1}^{n} \hat{\boldsymbol{w}}(i)/||\hat{\boldsymbol{w}}(i)||}{||\sum_{i=1}^{n} \hat{\boldsymbol{w}}(i)/||\hat{\boldsymbol{w}}(i)||||} & \text{if } \sum_{i=1}^{n} \hat{\boldsymbol{w}}(i)/||\hat{\boldsymbol{w}}(i)|| \neq 0 \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{42}$$

After the convergence of the LBG clustering algorithm, we obtain E clusters of i-vectors with their centroids denoted as $\boldsymbol{c}_1^{\hat{\boldsymbol{w}}}, \boldsymbol{c}_2^{\hat{\boldsymbol{w}}}, \ldots, \boldsymbol{c}_E^{\hat{\boldsymbol{w}}}$, respectively. We use $\boldsymbol{c}_0^{\hat{\boldsymbol{w}}}$ to denote the centroid of all the training i-vectors.

References

[1] F. Beaufays, V. Vanhoucke, and B. Strope, "Unsupervised discovery and training of maximally dissimilar cluster models," *Proc. Interspeech2010*, pp. 66-69.

[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19, No. 4, pp. 788-798, 2011.

[3] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. on Speech and Audio Processing*, Vol. 13, No. 3, pp. 345-354, 2005.

[4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communication*, Vol. COM-28, pp. 84-95, 1980.

[5] J. Xu, Y. Zhang, Z.-J. Yan, and Q. Huo, "An i-Vector based Approach to Acoustic Sniffing for Irrelevant Variability Normalization based Acoustic Model Training and Speech Recognition," *Proc. Interspeech2011.*

[6] S. Young, *et al.*, The HTK Book (for HTK version 3.4), 2006.

[7] Y. Zhang, J. Xu, Z.-J. Yan, and Q. Huo, "An i-vector based approach to training data clustering for improved speech recognition," *Proc. Interspeech2011*.

A The Matrix Codebook for this memo

A.1 Trace

$$\mathrm{tr}(AB) = \mathrm{tr}(BA) \tag{43}$$

$$\partial \mathrm{tr}(X) = \mathrm{tr}(\partial X) \tag{44}$$

$$\frac{\partial}{\partial X}\mathrm{tr}(AX^\top) = A \tag{45}$$

$$\frac{\partial}{\partial X}\mathrm{tr}(AXBX^\top) = AXB + A^\top XB^\top \tag{46}$$

$$\frac{\partial}{\partial X}\mathrm{tr}(AX^{-1}B) = -X^{-\top}A^\top B\top X^{-\top} \tag{47}$$

$$\frac{\partial}{\partial X}\ln|\det(X)| = (X^{-1})^\top \tag{48}$$

A.2 Variance of quadratic form

Assume that $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{R})$ and A is positive semi-definite,

$$E(\boldsymbol{x}\boldsymbol{x}^\top) = \boldsymbol{R} + \boldsymbol{\mu}\boldsymbol{\mu}^\top \tag{49}$$

$$E(\boldsymbol{x}^\top A\boldsymbol{x}) = \mathrm{tr}(A\boldsymbol{R}) + \boldsymbol{\mu}^\top A\boldsymbol{\mu} \tag{50}$$

Proof:

$\because$ A is positive semi-definite

$\therefore$ $A = C^\top C$

$\therefore$ $E(C^\top \boldsymbol{x}^\top \boldsymbol{x}C) = C^\top(\boldsymbol{R} + \mu\mu^\top)C$

$\therefore$ $E(\boldsymbol{x}^\top A\boldsymbol{x}) = E(\sum_i (C\boldsymbol{x})_i^2) = \mathrm{tr}(C^\top(\boldsymbol{R} + \mu\mu^\top)C) = \mathrm{tr}(AE(\boldsymbol{x}\boldsymbol{x}^\top))$

Note that

$$\begin{aligned}\mathrm{tr}(A\boldsymbol{\mu}\boldsymbol{\mu}^\top) &= \sum_i \sum_j a_{ij}\mu_i\mu_j \\ &= \sum_i A\boldsymbol{\mu}\mu_i \\ &= \boldsymbol{\mu}^\top A\boldsymbol{\mu}\end{aligned} \tag{51}$$

and let $\boldsymbol{\mu} = E(\boldsymbol{w}), \boldsymbol{R} = E(\boldsymbol{w}\boldsymbol{w}^\top) - E(\boldsymbol{w})E(\boldsymbol{w})^\top, A = \boldsymbol{T}^\top\Gamma(i)\boldsymbol{R}^{-1}\boldsymbol{T}$ we get Eq.[23].

A.3 Others

1 Proposition:

$$\boldsymbol{R}\Gamma(i) = \Gamma(i)\boldsymbol{R} \tag{52}$$

where

$$\boldsymbol{R} = \text{diag}\{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_C\} \tag{53}$$

$$\Gamma(i) = \text{diag}\{\gamma_1 \boldsymbol{I}, \ldots, \gamma_C \boldsymbol{I}\} \tag{54}$$

PROOF:

$$\begin{aligned} \boldsymbol{R}\Gamma(i) &= \text{diag}\{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_C\}\text{diag}\{\gamma_1 \boldsymbol{I}, \ldots, \gamma_C \boldsymbol{I}\} \\ &= \text{diag}\{\gamma_1 \boldsymbol{I}\boldsymbol{R}_1, \ldots, \gamma_C \boldsymbol{I}\boldsymbol{R}_C\} \\ &= \Gamma(i)\boldsymbol{R} \end{aligned} \tag{55}$$

In Mark J.F. Gales's paper, it assume that $\boldsymbol{w}$ is only a model parameter and find the solution through "the alternative of the variables method" (there is no closed-form maximum likelihood solution for $\boldsymbol{w}, \boldsymbol{T}, \boldsymbol{R}$.

B From a graphic model view

B.1 Probabilistic PCA

Probabilistic PCA is a simple example of the linear-Gaussian framework, in which all of the marginal and conditional distributions are Gaussian. We can formulate probabilistic PCA by first introducing an explicit latent variable w corresponding to the principal-component subspace. Next we define a Gaussian prior distribution $p(w)$ over the latent variable, together with a Gaussian conditional distribution $p(x|w)$ for the observed variable x conditioned on the value of the latent variable. Specifically, the prior distribution over w is given by a zero-mean unit-covariance Gaussian

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \boldsymbol{I}) \tag{56}$$

and the conditional distribution of the observed variable x

$$p(\boldsymbol{x}|\boldsymbol{w}) = \mathcal{N}(\boldsymbol{x}|T\boldsymbol{w} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}). \tag{57}$$

We can view the probabilistic PCA model from a generative viewpoint in which a sampled value of the observed variable is obtained by first choosing a value for latent variable and then sampling the observed variable conditioned on this latent value:

$$\boldsymbol{x} = T\boldsymbol{w} + \boldsymbol{\mu} + \epsilon \tag{58}$$

where w is an Gaussian latent variable, and ϵ is a zero-mean Gaussian-distributed noise variable with covariance $\sigma^2 \boldsymbol{I}$. The generative process is illustrated in Figure 2.

Figure 2: Probabilistic PCA

B.1.1 Maximum likelihood PCA