CSLT TECHNICAL REPORT-20160012 [Thursday 5th January, 2017]

基于Kaldi i-vector的说话人识别系统使用说明

Yixiang Chen[1]
, Lantian Li[1]
and Dong Wang[1*]

*Correspondence: wang-dong99@mails.tsinghua.edu.cn
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

受联合因子分析理论的启发，Dehak提出了从GMM均值超矢量中提取一个更紧凑的矢量，称为i-vector。这里的i是身份(Identity)的意思，出于自然的理解，i-vector相当于说话人的身份标识。本文叙述了基于Kaldi i-vector说话人识别系统的实现过程。

Keywords: i-vector; kaldi

1 介绍

说话人识别是一类典型的模式识别问题，包含说话人模型训练和测试语音打分判决两个阶段，

1. 训练阶段：对每个使用系统的说话人预留充足的语音；对预留语音提取声学特征；根据提取的声学特征训练得到说话人模型；将每个说话人模型存入说话人模型库中；

2. 测试阶段：系统获取待测试识别的语音；与训练阶段相同，提取测试语音的声学特征；将测试语音的声学特征与说话人模型库进行比对，根据预先定义的相似性准则，在说话人模型上进行打分判别；最后得到测试语音的说话人身份。

在说话人识别领域，目前使用的特征绝大部分是研究语音信号频率上短时倒谱(Short-term Cepstrum) 特性得到的声学特征，这些特征主要模拟语音信号中的底层声学特性，例如人耳的听觉特性、声道的发声机理等，主要就包括梅尔频率倒谱系数(Mel-Frequency Cepstral Coefficients，MFCC)。

在模型领域，基于统计的机器学习方法占据了主流地位，其中最经典的建模方法是基于高斯混合模型-通用背景模型(Gaussian Mixture Model-Universal Background Mode, GMM-UBM) [1, 2]的方法，如图1。由于不同说话人在声学特征上的差异，为此通过统计每个说话人的声学特征所具有的概率密度函数来构建说话人模型。高斯混合模型GMM是将空间的概率密度分布用多个高斯概率密度函数加权和来拟合，可以平滑地逼近任意形状的概率密度函数，并且是一个易于处理的参

数模型。在具体表示上，这个模型实际上就是把高斯混合模型的每个高斯分量的均值向量排列在一起组成一个超向量作为某一个说话人的模型，称为均值超矢量。可是在实际应用中，通常每一个说话人的语音数据很少，而训练高斯混合模型又需要大量的训练数据。于是，UBM通用背景模型被提了出来。在训练说话人模型时，利用一个预先训练好的与说话人无关的通用背景模型UBM和少量的说话人数据，通过相关自适应算法(如最大后验概率MAP [3]等)得到目标说话人模型。

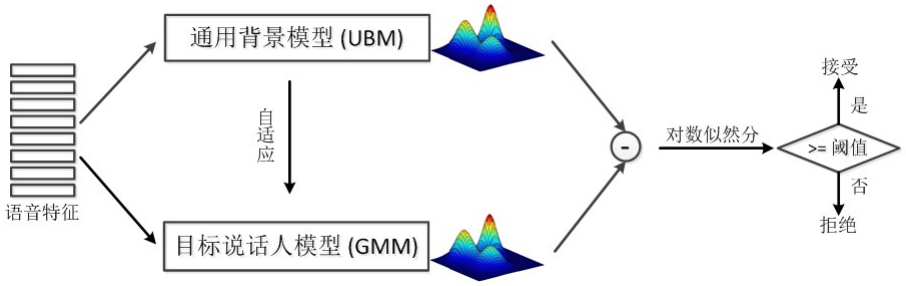

Figure 1 基于GMM-UBM的说话人识别系统流程图

上述的GMM-UBM系统一般作为说话人识别的基线系统。但是，该系统仍存在很多缺陷，例如其不能够很好的解决说话人识别的信道鲁棒性。因此，在GMM-UBM系统的基础上，基于因子分析的Joint Factor Analysis，JFA [4] 和i-vector [5]模型的说话人识别系统应运而生。

传统的联合因子分析JFA建模过程主要是基于两个不同的空间：由本征音空间矩阵定义的说话人空间和由本征信道空间矩阵定义的信道空间。受联合因子分析理论的启发，Dehak提出了从GMM均值超矢量中提取一个更紧凑的矢量，称为i-vector。这里的I是身份(Identity)的意思，出于自然的理解，i-vector相当于说话人的身份标识。与JFA不同，i-vector方法采用一个空间来代替这两个联合因子空间(说话人空间和信道空间)，这个新的空间称为全局变量空间，它即包含了说话者之间的差异又包含了信道间的差异。因此，i-vector在建模过程中并不严格区分说话者和信道。这一建模方法的动机来源于Dehak的又一研究: JFA建模后的信道因子不仅包含了信道信息，也夹杂着说话人的信息 [5]。

因子分析是i-vector模型的理论支持，因此在讲述i-vector模型之前，首先需要了解一下因子分析。因子分析(factor analysis) 是一种数据简化的技术。它通过研究众多变量之间的内部依赖关系，探求观测数据中的基本结构，并用少数几个假想变量来表示其基本的数据结构。对于m 个n 维特征的训练样例$x^{(i)}(x_1^{(i)}, x_2^{(i)}, ..., x_n^{(i)})$的产生过程如下：

1、首先在一个k维的空间中按照多元高斯分布生成m个$z^{(i)}$(k维向量)，即$z^{(i)}$ $N(0,1)$。

2、然后存在一个变换矩阵$\Lambda \in \mathbb{R}^{n\times k}$，将$z^{(i)}$映射到n维空间中，即$\Lambda z^{(i)}$因为$z^{(i)}$的均值是0，映射后仍然是0。

3、然后将$\Lambda z^{(i)}$加上一个均值μ(n维)，即$\mu + \Lambda z^{(i)}$ 对应的意义是将变换后的$\Lambda z^{(i)}$(n维向量)移动到样本$x^{(i)}$的中心点μ。

4、由于真实样例$x^{(i)}$与上述模型生成的有误差，因此我们继续加上误差ϵ(n维向量)，而且ϵ符合多元高斯分布，即$\epsilon\ N(0,\Psi)$。

5、最后的结果认为是真实的训练样例$x^{(i)}$的生成公式

$$x^{(i)} = \mu + \Lambda z^{(i)} + \epsilon$$

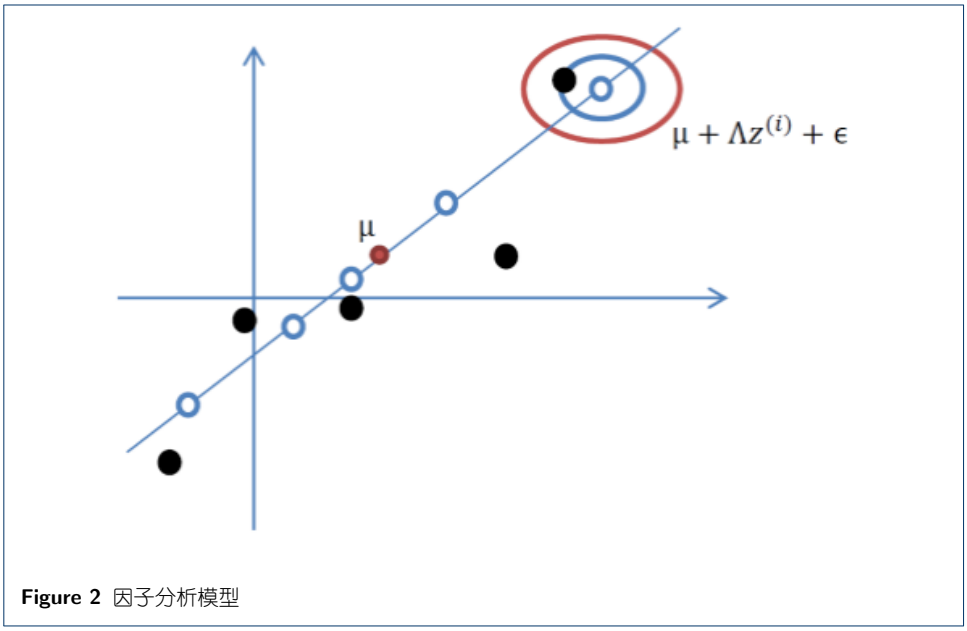

Figure 2 因子分析模型

因子分析理论如图2。其实，i-vector模型就是最基本的因子分析模型。在i-vector系统中，通用背景模型(UBM)的高斯均值超向量对应于因子分析中的μ，全变量空间(T-matrix)对应于因子分析中的因子变换矩阵Λ。因此i-vector模型可表示为：

$$M = m + T\tilde{\mathbf{w}} \tag{1}$$

基于i-vector的说话人识别系统如图3所示。它是基于单一全变量空间的跨信道算法，该全变量空间中既包含了说话人空间信息也包含了信道空间信息。i-vector模型相当于采用因子分析方法将高维语音空间投影到低维子空间中。

在进行说话人确认时，首先对待确认语音段X_v 和待匹配语音段X_e 分别计算其i-vector $\tilde{\mathbf{w}}_v$和$\tilde{\mathbf{w}}_e$, 再计算两者的余弦距离得到匹配分数：

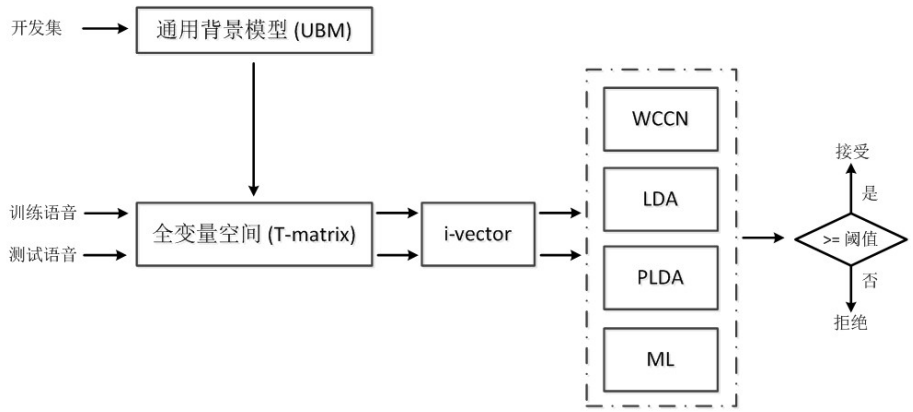

Figure 3 基于i-vector模型的说话人识别系统流程图

$$S_{v,e} = \frac{< \tilde{\mathbf{w}}_v, \tilde{\mathbf{w}}_e >}{\sqrt{||\tilde{\mathbf{w}}_v|| ||\tilde{\mathbf{w}}_e||}} \tag{2}$$

由于i-vector模型中既包含说话人信息又包含信道信息，而说话人识别任务仅关心如何提取准确鲁棒的说话人信息。因此，为了尽可能的减小信道信息对于说话人识别产生的干扰，基于i-vector模型陆续提出一系列信道补偿的算法，其中最为常用的信道补偿算法是LDA (Linear discriminative analysis) 和PLDA (Probabilistic linear discriminative analysis) [6]算法。

首先简要的介绍一下LDA模型。i-vector模型训练和计算中并未考虑同一说话人内部的变化性，因此得到的i-vector向量既表征了说话人信息，也表征了语音内容、信道、情绪等说话人内部变化信息。为进一步区分说话人之间与说话人内部的变动性，研究者提出采用LDA方法，通过一个线性映射矩阵，在对i-vector进一步降维的同时突出说话人之间的差异性。设第i个人的第j个i-vector 为$\tilde{\mathbf{w}}(i,j)$, LDA方法目标函数如下:

$$J(G) = \frac{S_B}{S_W} \tag{3}$$

其中：

$$\begin{aligned} S_B &= \sum_i (\mu_i - \mu)^2 \\ S_W &= \sum_i \{\sum_j (\mathbf{w}'(i,j) - \mu_i)^2\} \\ \mathbf{w}'(i,j) &= G\tilde{\mathbf{w}}(i,j) \end{aligned}$$

上述模型中$G \in R^{K \times M}$为i-vector的LDA映射矩阵，K 为LDA 所映射到的低维空间的维度。μ_i 为第i个说话人的i-vector在映射空间上的均值，μ 为所有说话人的i-vector 在映射空间上的均值。通过优化G，LDA可以将i-vector向量映射到低维空间，同时保证在映射空间里不同说话人之间的区分性在公式(3)所定义的准则上最大化。经过LDA降维后，说话人确认可以在低维映射空间上依公式(2) 实现。

LDA方法可以扩展为一个产生式PLDA模型 [6]，具有更好的信道补偿能力。假设训练数据语音有I个说话人，其中每个说话人有J条语音。那么，我们定义第i个说话人的第j条语音为X_{ij}。然后，根据因子分析，我们定义X_{ij}的生成模型为：

$$X_{ij} = \mu + Fh_i + Gw_{ij} + \epsilon_{ij} \tag{4}$$

这个模型可以看成两个部分：等号右边前两项只跟说话人有关而跟说话人的具体某一条语音无关，称为信号部分，这描述了说话人类间的差异；等号右边后两项描述了同一说话人的不同语音之间的差异，称为噪音部分。这样，我们用了这样两个因子变量来描述一条语音的数据结构。

我们注意到等号右边的中间两项分别是一个矩阵和一个向量的表示形式，这便是因子分析的表达形式。这两个矩阵F和G包含了各自假想变量空间中的基本因子，这些因子可以看做是各自空间的特征向量。比如，F的每一列就相当于类间空间的特征向量，G的每一列相当于类内空间的特征向量。而两个向量可以看做是分别在各自空间的特征表示，比如h_i就可以看做是X_{ij} 在说话人空间中的特征表示。在测试阶段，我们不再像LDA那样去基于cosine距离来计算得分，而是去计算两条语音是否由说话人空间中的特征h_i生成，或者由h_i生成的似然程度。在这里，我们使用对数似然比来计算得分。这样说话人确认就变成了个假设检验问题。即如果有两条测试语音，这两条语音来自同一空间的假设为H_s，来自不同的空间的假设为H_d，那么通过计算对数似然比，就能衡量两条语音的相似程度。得分越高，则两条语音属于同一说话人的可能性越大。如下公式所示：

$$Score = log\frac{p(\tilde{\mathbf{w}}_v, \tilde{\mathbf{w}}_e|H_s)}{p(\tilde{\mathbf{w}}_v|H_d)p(\tilde{\mathbf{w}}_e|H_d)} \tag{5}$$

2 实验

在Kaldi工具包中，其提供了基于GMM i-vector说话人识别系统的标准脚本，默认位于Kaldi-path/egs/sre08/v1下。基于GMM i-vector说话人识别系统的源代码位于Kaldi-path/src下，常用的目录有feat、featbin(特征提取) 和ivector、ivectorbin(模型训练和打分测试)。

基于Kaldi的GMM i-vector说话人识别系统主要由四个步骤组成：数据列表准备(scp文件)、特征提取(MFCC提取及VAD)、模型训练(UBM、T-matrix训练

和i-vector提取)、打分判决(Cosine、LDA、PLDA)，其流程示意图及所调用的相关指令如图 4所示。

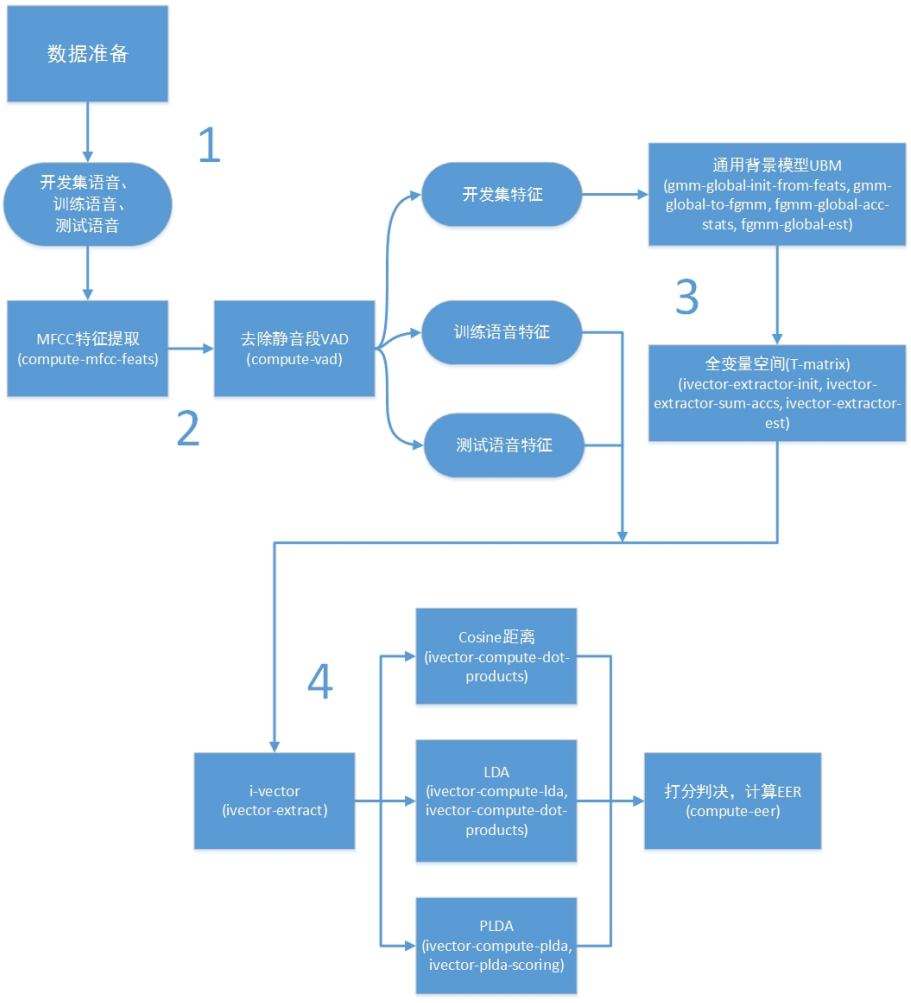

Figure 4 基于Kaldi的GMM i-vector说话人识别系统流程示意图

2.1 数据列表准备(scp文件)

对开发集语音、训练集语音和测试集语音分别生成对应的数据文件。最基本的数据文件包括：wav.scp (语音句子标签与其语音存放路径)；spk2utt (说话人标签与语音句子标签的对应关系)；utt2spk (语音句子标签与说话人标签的对应关系)。

首先抓取语音数据集中的全部语音文件('.wav' ， '.sph'等)，生成每句话及其对应路径的wav.scp文件。wav.scp文件中有两列，第一列为语音句子标签，第二列为语音所在路径。其格式如下：

utt_1 $path$/utt_1.wav

utt_2 $path$/utt_1.wav

... ...

注：wav.scp文件每一行的输出流必须是8KHz 或16KHz 的'.wav'格式。因此，若原始语音数据为'.sph'，则需用sph2pipe指令进行格式转换；若原始语音为非8KHz 或16KHz 的'.wav'格式，则需用sox指令进行格式转换。

根据wav.scp文件，准备utt2spk文件。utt2spk文件也有两列，第一列为语音句子标签，第二列为语音所对应说话人的标签。其格式如下：

utt_1 spk_A

utt_2 spk_A

utt_3 spk_B

utt_4 spk_B

... ...

注：对于开发集语音和训练集语音，其utt2spk数据格式如上。而对于测试集语音中，由于测试时并不知道每条语音所对应的说话人，所以在其utt2spk的文件中，假设每句话即为一个说话人，因此utt2spk的第二列语音所对应说话人的标签就是第一列的语音句子标签。其格式如下：

utt_1 utt_1

utt_2 utt_2

utt_3 utt_3

utt_4 utt_4

... ...

准备好wav.scp文件和utt2spk文件后，使用$utils/fix_data_dir.sh$ 脚本对指定路径下所有文件按照Kaldi格式排序，命令如下：

$sh \quad utils/fix\_data\_dir.sh \quad path$

然后调用$utils/utt2spk_to_spk2utt.pl$脚本利用utt2spk文件列表生成spk2utt文件列表，命令如下：

$perl \quad utils/utt2spk\_to\_spk2utt.pl \quad utt2spk > spk2utt$

注：准备好的spk2utt文件行数应该与说话人数量相等，每行第一列为说话人标签，其后跟N列是该说话人所说的语音标签，其格式如下：

spk_A utt_1 utt_2 ...

spk_B utt_3 utt_4 ...

... ...

根据上述步骤，对开发集语音、训练集语音和测试集语音分别准备好对应的数据文件，并存至对应的文件夹中。

此外，根据训练集和测试集生成一个评测系统性能的测试列表trials。通常我们采用全交叉的打分方式，因此trials的长度为‘训练集中说话人的个数’ * ‘测试集中语音句子的个数’。该trials的每行有三列，第一列为测试集中语音句子标签，第二列为训练集中说话人标签；第三列有两种标签，当前两列为真实对应关系(即第一列的测试语音是由第二列的说话人所说)，则第三列标记为‘target’；反之为‘nontarget’。示例如下：

utt_1	spk_A	target
utt_2	spk_B	nontarget
utt_3	spk_A	nontarget
utt_4	spk_B	target
...	...	...

2.2 特征提取(MFCC提取及VAD)

根据准备好的数据列表，依次对开发集、训练集和测试集的语音进行特征提取。在Kaldi中，默认是采用梅尔频率倒谱系数(Mel-Frequency Cepstral Coefficients，MFCC)作为说话人识别任务的声学特征。

其中MFCC特征的提取过程如图 5所示：

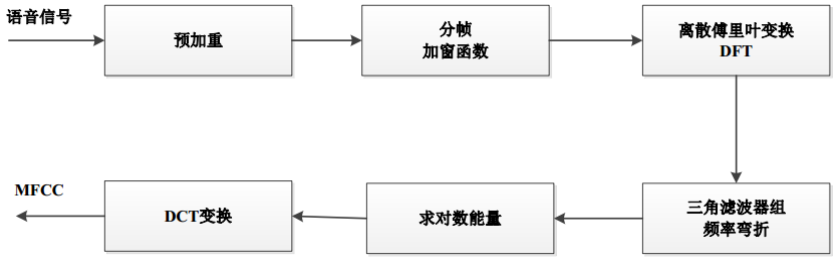

Figure 5 MFCC特征提取流程

Kaldi封装了一个MFCC特征提取的脚本，可实现分布式运算，详情参阅*steps/make_mfcc.sh*。该脚本中的核心指令是compute-mfcc-feats，该指令用于MFCC特征的计算，其用法如下：

compute-mfcc-feats [options...] <wav-rspecifier><feats-wspecifier>

其中<wav-rspecifier>为数据目录下的wav.scp文件；<feats-wspecifier>为生成的feats.scp 文件。

[options...]是compute-mfcc-feats的一些特征参数，其将决定提取MFCC特征的数值。常用的特征参数如下：

–sample-frequency	语音数据的采样率
–frame-length	每帧的时长(决定计算FFT点的个数)
–high-freq	高频上界
–low-freq	低频下界
–num-mel-bins	三角滤波器的个数
–num-ceps	DCT运算后的维度，即为MFCC特征维度
–use-energy	提取每帧的语音能量，用于Energy-VAD

对于一条语音，其通常是由若干语音段组成，而段与段之间为静音段，即在该段内说话人并未发音。这些静音段对说话人识别任务没有意义，因此需要采用有效音检测(Voice activity detection, VAD)的方法，把语音中说话人发音的语音段提取出来，而未发音的静音段剔除。在Kaldi中，调用*sid/compute_vad_decision.sh*对语音实现VAD。脚本中最重要的指令是compute-vad，该指令通过feats.scp文件生成对应的vad.scp文件。

compute-vad [options] <feats-rspecifier >< vad-wspecifier >

系统中的VAD采用基于能量的检测方法，即当语音帧能量值低于设定阈值时，认定其为静音段；反之为有效音段。compute-vad常用的[options]为–vad-energy-mean-scale 和–vad-energy-threshold，由其控制能量阈值的选取。

2.3 模型训练(UBM、T-matrix训练和i-vector提取)

由图 4可以看出，i-vector模型训练过程主要分为三个部分：通用背景模型UBM训练，T 矩阵训练和i-vector提取。

UBM的训练过程分为两步，首先从开发集中随机抽取少量语音数据，训练一个对角-协方差的UBM，该过程基于脚本*sid/train_diag_ubm.sh*; 然后再使用*sid/train_full_ubm.sh*训练全角-协方差的UBM。

在UBM训练完成后，利用UBM计算每句话的充分统计量，用于T矩阵的训练。该过程调用脚本*sid/train_ivector_extractor.sh*，并利用全部开发集的数据通过迭代训练得到T矩阵。

在T矩阵训练完成后，可调用*sid/extract_ivectors.sh*完成对语音数据的i-vector 提取。该脚本的功能是给定UBM、T矩阵和相关语音特征文件(feats.scp, vad.scp)，输出语音特征文件对应的i-vector模型。主要步骤如下：

a. Set various variables：将语音特征文件划分为n等份；(执行n个jobs，并行加速运算)

b. Set up features: 对语音特征进行VAD，CMVN和一阶/二阶差分处理。

c. Extracting i-vectors：对处理后的语音特征计算对应的i-vector，具体步骤如下：

1). fgmm-global-to-gmm (Convert single full-covariance GMM to single diagonal-covariance GMM.)

2). gmm-gselect (For each frame, gives a list of the n best Gaussian indices, sorted from best to worst.)

3). fgmm-global-gselect-to-past (Given features and Gaussian-selection (gselect) information for a full-covariance GMM, output per-frame posteriors for the selected indices.)

4). scale-post (Scale posteriors with either a global scale, or a different scale for each utterance.)

5). ivector-extract (Extract iVectors for utterances, using a trained iVector extractor, and features and Gaussian-level posteriors.)

d. Combing i-vectors across jobs: 将n个jobs输出的i-vectors合并。

其中，指令ivector-extract是利用已训练好的i-vector extractor(T-matrix)、语音特征feats.scp和UBM混合的后验概率posteriors，为每句话提取i-vector。其使用方法如下

ivector-extract [options] <model-in><feature-rspecifier><posteriors-rspecifier><ivector-wspecifier>

2.4 打分判决(Cosine、LDA、PLDA)

在识别测试阶段，首先提取训练集和测试集语音的i-vectors。对于训练集，通过指令ivector-mean和参数列表spk2utt，提取每个说话人的*spk_ivector*模型；并利用指令ivector-normalize-length完成*spk_ivector*的二阶norm，最后存入*spk_ivector.ark* 和*spk_ivector.scp*中。对于测试集，指令ivector-normalize-length完成测试语音*ivector*的二阶norm。

在得到训练集每个说话人模型列表*spk_ivector.scp*和测试集每条语音的模型列表*ivector.scp*后，利用预先准备的测试列表trials，即可完成相关测试任务。需要注意的是，Kaldi中采用等错误率(equal error rate, EER)作为说话人识别系统性能的评价指标。EER 为错误接受率（False Alarm Rate，FAR）和错误拒绝率（False Rejection Rate，FRR）这两个值相等时的点。FAR 为错误接受假冒闯入者的比率，一般代表系统的安全性，这个值越低系统越安全；FRR 为错误拒绝正确说话人的比率，一般代表系统接受说话人的容易程度，这个值越低越容易接受当前说

话人。检测错误权衡曲线（Detection Error Trade-offs Curve，DET Curve）就是以FAR 为横轴，FRR 为纵轴的性能曲线，而这条曲线中FAR 和FRR 相等的点即为EER。通常希望系统的EER 尽量低，即FAR 和FRR 同时都尽可能更小。FAR 和FRR 都是受到系统阈值的影响，一般阈值升高的时候FRR 升高FAR 降低，反之降低时FRR 变低FAR 升高。所以用EER 来描述了说话人确认系统的一个平均性能。

Kaldi提供了三种识别打分方法，Cosine，LDA和PLDA打分。

1、Cosine打分：该方法最为简单快捷，使用指令ivector-compute-dot-products计算测试语音i-vector 与对应说话人模型i-vector之间的Cosine距离，作为判决分数。指令ivector-compute-dot-products按照trials文件对应顺序计算trials文件每对测试语音i-vector的向量点积，使用格式为：

ivector-compute-dot-products [options] <trials-in><ivector1-rspecifier><ivector2-rspecifier><scores-out>

2、LDA打分：首先利用开发集，使用指令ivector-compute-lda训练LDA矩阵，指令格式为：

ivector-compute-lda [options] <ivector-rspecifier><utt2spk-rspecifier><lda-matrix-out>

将测试语音i-vector与对应说话人模型i-vector经过LDA矩阵降维，而后对降维的i-vectors利用ivector-compute-dot-products计算其之间的距离，作为判决分数。

3、PLDA打分：首先利用开发集，使用指令ivector-compute-plda训练PLDA模型，再通过ivector-plda-scoring计算最大似然比得到判决分数。指令格式为:

ivector-compute-plda [options] <spk2utt-rspecifier><ivector-rspecifier><plda-out>

ivector-plda-scoring <plda><train-ivector-rspecifier><test-ivector-rspecifier><trials-rxfilename><scores-wxfilename>

在得到打分文件<scores-wxfilename>后，调用compute-eer完成等错误率EER的计算。指令compute-eer的输入为一个列表文件；每行两列，第一列是置信分数，第二列代表是自识别分数还是闯入识别分数。计算EER的实例脚本可参照$local/score_sre08.sh$。

3 补充内容DNN i-vector模型

Kaldi中，除了上述的GMM i-vector说话人识别系统外，还提供了DNN i-vector [7, 8]说话人识别系统。其基本思想是利用基于深度神经网络的语音识别模型DNN-ASR替换通过非监督聚类的高斯混合模型UBM，如图 6所示。

Kaldi中DNN i-vector的脚本位于$egs/sre10/v2/run.sh$。该脚本提供了两种DNN i-vector的方法。第一，利用DNN-ASR训练有监督的UBM，即为UBM-

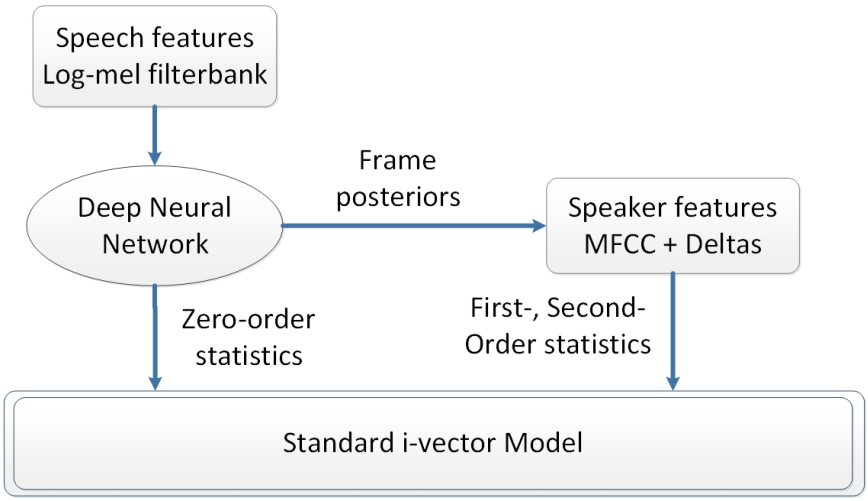

Figure 6 DNN i-vector模型

sup；然后用UBM-sup替换原始非监督聚类得到的UBM训练T矩阵；而后过程与GMM i-vector一致；第二，利用DNN-ASR训练有监督的UBM和T矩阵；i-vector提取是基于有监督的UBM和T矩阵。

图 7给出了GMM i-vector 和DNN i-vector模型的不同之处。

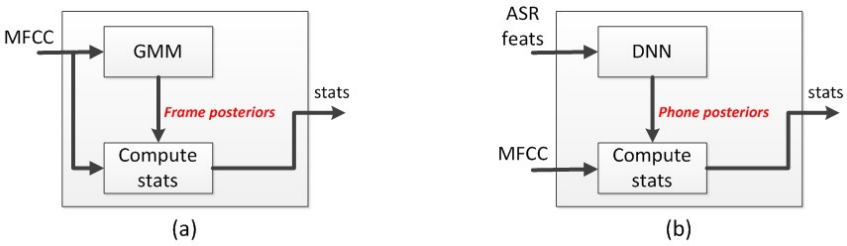

Figure 7 GMM i-vector 与DNN i-vector 对比示意图

总结一下，两者主要有三点不同。

1、用于计算充分统计量的后验概率不同：GMM i-vector的后验概率是每一帧在每个高斯混合上的概率；而DNN i-vector的后验概率是每一帧经过DNN-ASR解码得到的在每个音素(senones)上的概率，即为语音识别的音素后验概率。

2、T矩阵中每个子空间的意义不同：GMM i-vector中，每个高斯混合代表一个类别(region/class)；DNN i-vector中，DNN的每个输出结点(senones)代表一个音素子类(phonetic region/class)。

3、训练方式不同：GMM i-vector为基于EM算法的无监督学习；DNN i-vector为基于DNN的区分性有监督学习。

Acknowledgement

Author details

[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [2]Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [3]Department of Computer Science and Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China.

References

1. Douglas A Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech communication*, vol. 17, no. 1, pp. 91–108, 1995.
2. Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.
3. Jean-Luc. Gauvain and Chin-Hui. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.
4. Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.
5. Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
6. Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *ICCV'07*. IEEE, 2007, pp. 1–8.
7. Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Moray McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1695–1699.
8. David Snyder, Daniel Garcia-Romero, and Daniel Povey, "Time delay deep neural network-based universal background models for speaker recognitio," *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 92–97, 2015.