

iVector-based

iVector/PLDA, 最佳基线模型

DNN-based

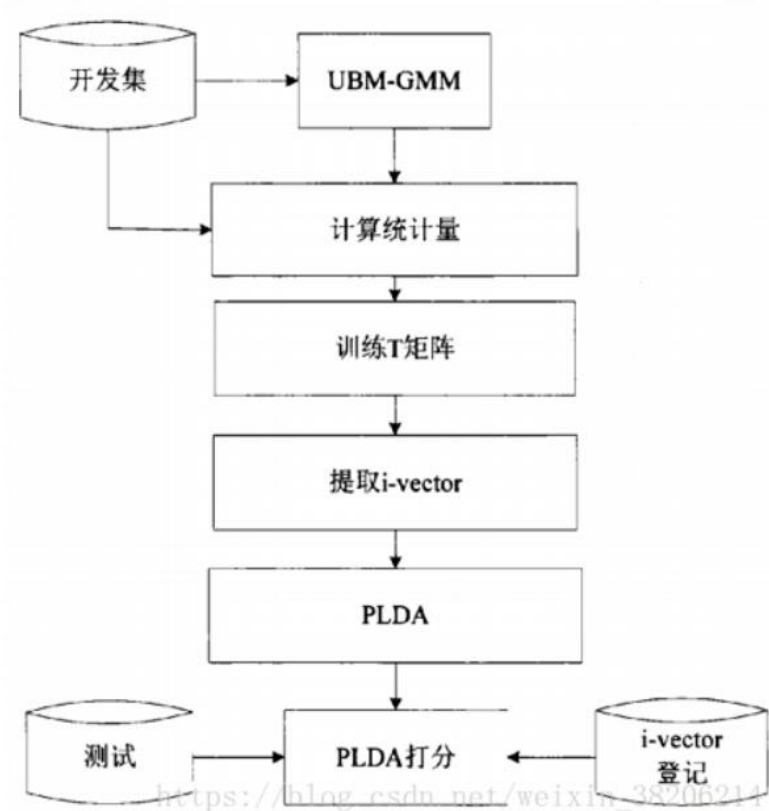
- ✓ x-vector, Bottleneck Feature, d-vector, j-vector 等, 基本思想都是从神经网络中提取某层的权重作为说话人身份向量。然后有两种做法: 1) 直接计算 vector 之间的 Cosine Distance; 2) 或者再加个 back-end 分类器。x-vector, d-vector, j-vector, BNF 等都可以归纳到 embeddings 类型上。
- ✓ End-to-End, 即给定 Speech, 直接从网络给出拒绝或接受。

attention mechanism、  
Learning to rank 等改良  
类

- ✓ Attention mechanism 在 CV 上的效果非常优秀, Google 在 Speech Recognition 和 Speaker Recognition 上做了有效应用 [10]。
- ✓ Learning to rank 也是被 Google 用于 Face recognition 取得不错的效果, 被借鉴用来做 Speaker recognition 的。Triplet Loss 就是 Learning to rank 步骤中的一个 similarity function [6][15]

# Speaker Verification Using Adapted Gaussian Mixture Models

## Gaussian Mixture Model-Universal Background Model (GMM-UBM)



Assuming  $R$  utterances for a speaker, the collection of corresponding i-vectors is denoted as  $\{\eta_r: r = 1, \dots, R\}$ . The G-PLDA model introduced in [3] then assumes that each i-vector can be decomposed as

$$\eta_r = m + \Phi\beta + \Gamma\alpha_r + \epsilon_r. \quad (2)$$

在传统的基于GMM-UBM的识别系统中，由于训练环境和测试环境的失配问题，会导致系统性能不稳定。联合因子分析(Joint Factor Analysis, JFA)认为，说话人的GMM模型的差异信息，是由说话人差异和信道差异这两个不可观测的部分组成的，公式如下：

$$M = s + c$$

其中， $s$ 为说话人相关的超矢量，表示说话人之间的差异；

$c$ 为信道相关的超矢量，表示同一个说话人不同语音段的差异；

Linux 查看当前文件夹下所有目录的大小 `du -sh *`

Linux查看服务器已使用空间大小 `df -hl`

MobaXterm sh pycharm.sh 本地打开远程pycharm

Anaconda查看所有环境 `conda info -env`

修复pip不可用, bad interpreter: No such file or directory  
<https://blog.csdn.net/ganxiwu9686/article/details/98736127>

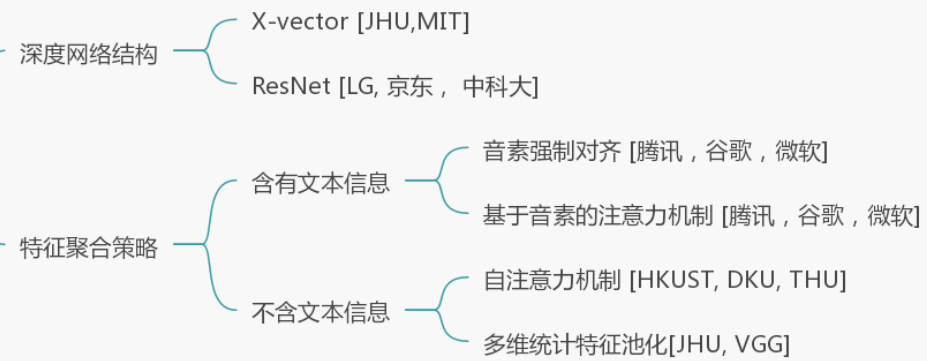
同, 围绕这一问题, Kenny 提出了联合因子分析 (Joint Factor Analysis, JFA) 技术为信道失配环境下的声纹识别研究开辟了新思路, 其主要思想是将说话人高斯均值超矢量所在空间划分为三个组成部分: 本征信道 (Eigenchannel) 空间、本征音 (Eigenvoice) 空间和残差 (Diagonal Residual) 空间, 通过移除说话人均值超矢量在本征信道空间的影响, 来达到抗信道失配的目的。然而, 在各种信道下训练数据不均衡时, JFA

**Table 3.** Comparison between *CN-Celeb* and *VoxCeleb*.

	<i>CN-Celeb</i>	<i>VoxCeleb</i>
Source media	bilibili.com	youtube.com
Language	Chinese	Mostly English
Genre	11	Mostly interview
# of Spks	1,000	7,363
# of Utters	130,109	1281,762
# of Hours	274	2,794
Human Check	Yes	No

# 短语语音说话人识别

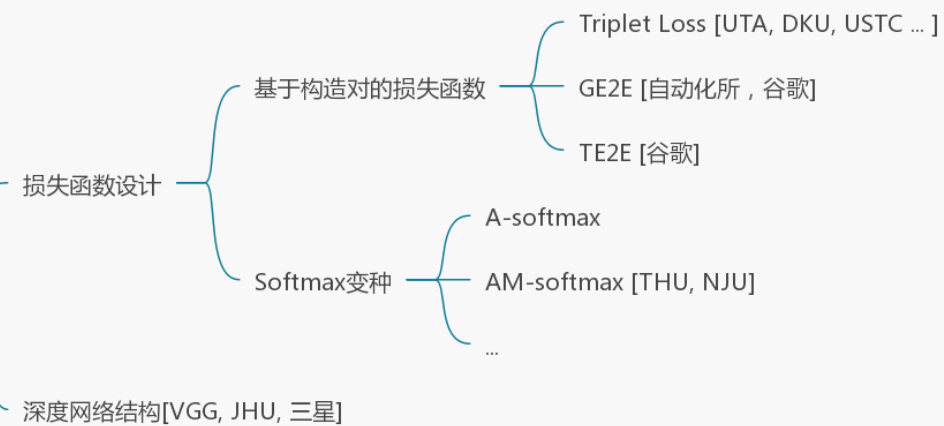
## 特征提取器设计



## 后端分类器设计



## 端到端系统



Linux 中断进程 kill PID / kill -s 9 PID  
tar -zxvf 压缩文件名.tar.gz

Kaldi 由utt2spk生成spk2utt  
utils/utt2spk\_to\_spk2utt.pl data/train/utt2spk > data/train/spk2utt

```
import random
print(random.randint(0,9))
```

创建软链接  
ln -s /work/pjy/Emotion/nn /work/pjy/nn

Jupyter notebook 使用已存在的anaconda环境  
python -m ipykernel install --user --name 环境名称 --display-name "Python (环境名称)"

根据PID查看进程信息  
ll /proc/PID

查看各用户占用内存  
ps aux | awk 'NR!=1{a[\$1]+=\$6;} END { for(i in a) print i " ", " a[i]"KB";}'

Awk使用环境变量  
awk -v val=\$x '{print \$1, \$2, \$3, \$4+val, \$5+ENVIRON["y"]}' OFS="\t" score.txt

输出重定向(覆盖)      输出重定向(追加)  
who > a.txt            who >> a.txt

chmod 777 a.txt (所有人都可以rwx)

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
<b>d</b>	<b>rwX</b>			<b>r-X</b>			<b>r-X</b>		
目录 文件	读	写	执行	读	写	执行	读	写	执行

Touch 创建文件

pip安装python库方式  
Pip3 install --user gpustat

Pwd 显示当前目录

打开服务器sublime  
Subl

Cp 源 目的

-c 指定channel

Cp -r 复制文件夹

Mv 源 目的

结束进程  
Kill -9 PID

Rm 删除

查看文件夹大小  
du -h --max-depth=1

LI 列出目录及其参数

../ 上级目录

给目录下所有文件加上777权限  
chmod -R 777 ~/tf-kaldi-speaker/egs/voxceleb/v1

./ 本级目录

~ 等价于本用户目录

Linux下编译运行脚本的时候出现"/bin/bash^M:  
没有那个文件或目录(bad interpreter: No such file or directory)"这样的错误  
使用在终端输入sed -i 's/\r\$//' check\_tool.sh即可

Cat 查看文件内容

查看系统硬盘空间  
df -h