MPhil in Machine Learning and Machine Intelligence

Module MLMI2: Speech Recognition

L12: Acoustic Adaptation, Adaptive Training & Speaker Embeddings

Phil Woodland
pcw@eng.cam.ac.uk

Michaelmas 2021

Cambridge University Engineering Department

Introduction

This lecture concentrates on acoustic adaptation.

We will mainly discuss adaptation to **speaker** differences, but also many of the same techniques can be applied to adaptation to **acoustic environment** changes.

Initially we will concentrate on methods for GMM-HMMs, but note that some of these techniques can be used with DNN-HMMs also.

We will include

- What is speaker adaptation? & causes of speaker variations
- Modes of adaptation
- feature normalisation and vocal tract length normalisation
- Speaker clustering & eigenvoices
- MAP adaptation
- Linear regression (MLLR and CMLLR)
- Adaptive Training
- Techniques for DNN adaptation

We will also briefly discuss extraction of **speaker embeddings** and their use in **diarisation**

Speaker Adaptation

Aim is to take an initial, well trained, GMM-HMM or DNN-HMM model and use data from a new speaker, the **adaptation data**, to improve the performance on the new speaker.

An adaptation scheme is good if

1. given sufficient data, tends towards speaker dependent performance
2. is effective with small amounts of data (**rapid** adaptation)

A variety of transforms and mapping schemes have been examined for speaker adaptation and many of these can be applied also to acoustic environment adaptation

- also environment-adaptation specific techniques not covered here

Sometimes a distinction is made between adapting the model parameters and adapting the features to all appear similar (**feature/speaker normalisation**).

Speaker differences can be divided into

- **Linguistic/Accent Differences**: regional accent, native/non-native speaker, pronunciations
- **Physiological Differences**: male/female, fundamental frequency, vocal tract length etc

In addition to inter-speaker variations, may also need to adapt to intra-speaker variations.

- Style differences e.g. formal vs casual.
- Transitory effects e.g. having a cold.

These effects increase the within speaker, **intra-speaker**, variability.

Speaker Adaptation Approaches and Adaptive Modes

There are different types of approaches

- feature (or speaker) normalisation: make features (more) speaker independent
- speaker clustering: use appropriate model set
- MAP-type approaches estimated using some adaptation data
- linear transforms of model parameters (or features) estimated using adaptation data
- for DNNs, can use of additional input features describing speaker

The **mode** of adaptation depends on the task being investigated

- incremental: results are required causally, the adaptation data is not all available in one block - dictation tasks, car navigation
- batch: all the data is available (or can be used) in one block: e.g. off-line transcription

In addition for batch adaptation the adaptation data may be

- supervised: the correct transcription of the adaptation data is known (dictation enrolment)
- unsupervised: no transcribed adaptation data available, transcription must be hypothesised (off-line transcription)

In general we are trying to apply adaptation techniques so that

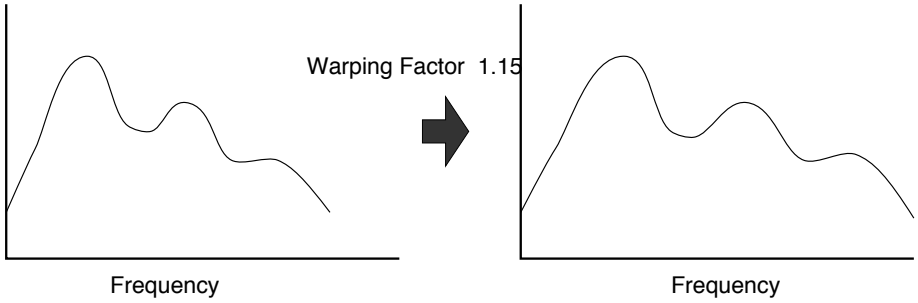
- obtain the performance of a Speaker (or Environment) dependent system
  with orders-of-magnitude less data (30 seconds vs thousands of hours!)

Feature Normalisation & VTLN

In feature/speaker normalisation aim is to construct a normalised feature space into which speech from any speaker may be mapped, so that inter-speaker differences are minimised.

One technique is **Vocal-Tract Length Normalisation**

- "Length" of the vocal tract is estimated and spectral shifts calculated accordingly.
- Typically estimated by a **Discrete search**: Set of discrete warping factors are examined, e.g. in the range 0.88 to 1.12. The one yielding the maximum likelihood is selected.
- Apply by Filter-bank shifting: Alter centre frequency according to warping factor.



Note that also **Linear Cepstral Transformations** can be used

- General linear transformation of the feature-space is used, trained in an ML fashion.
- Can also approximate VTLN

Speaker Clustering

A number of model sets are generated: adaptation process is then to decide which model set the current speaker belongs to.

Speakers are therefore **clustered** into groups and models trained for these speaker groups.

The adaptation process then selects the speaker group most representative of the current speaker. This is now a standard clustering problem.

There are problems with this approach.

1. The spectral variation of speakers, even within a cluster, may be large.
2. The new speaker may be not well represented by the training set of speakers.
3. It is assumed that the differences between speakers are uniform across a model set.

To over come some of these problems a combination of clusters may be used.

Rather than selecting a single cluster (speaker clustering), each speaker,is described as a point in speaker space (often an **eigenspace** that represents the speaker variability).

- Since low-dimensional speaker space, few speaker specific model parameters

Eigenvoices

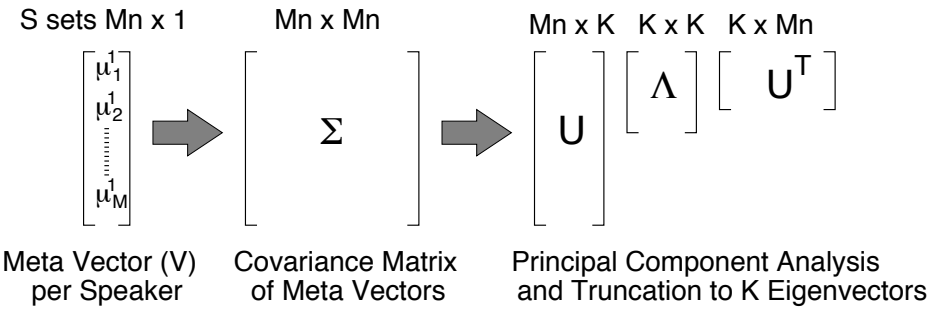
Generalisation of speaker clustering is "eigenvoices". Here some low-dimensional sub-space, the *eigenspace*, that models the most variability of the speakers is generated.

Each speaker, s, described as a point in eigenspace ($\boldsymbol{\lambda}^{(s)}$).

For an GMM-HMM, to simplify the implementation only means are adapted. Adapted mean may be written as

$$\hat{\boldsymbol{\mu}}_j^{(s)} = \sum_{i=1}^{K} \lambda_i^{(s)} \boldsymbol{\mu}_j^{(i)} = \left[\begin{array}{ccc} \boldsymbol{\mu}_j^{(1)} & \ldots & \boldsymbol{\mu}_j^{(K)} \end{array} \right] \boldsymbol{\lambda}^{(s)} = \boldsymbol{M}_j \boldsymbol{\lambda}^{(s)}$$

where $\boldsymbol{\mu}_j^{(k)}$ is the section of the k^{th} eigenspace dimension relating to component j.



The generation of the K dimensional *eigenspace* is shown above.

The stages are:

1. Meta-vector, $\boldsymbol{V}^{(s)}$, (all means spliced together) generated for each training speaker.
2. Covariance matrix of the set of meta-vectors is formed.
3. Principal component analysis of the covariance matrix. Eigenvectors associated with the largest K eigenvalues, $\boldsymbol{U}$, are selected as the eigenspace.

The position of a speaker in eigenspace, $\boldsymbol{\lambda}^{(s)}$, may be determined using:

1. **Projection**: for the new speaker, s, estimate the meta-vector if the means, $\boldsymbol{V}^{(s)}$. Project this vector into the eigenspace.

$$\boldsymbol{\lambda}^{(s)} = \boldsymbol{U}^T \boldsymbol{V}^{(s)}$$

2. **Maximum Likelihood**, an ML solution for the value of $\boldsymbol{\lambda}^{(s)}$ may also be estimated.

$$\boldsymbol{\lambda}^{(s)} = \arg\max_{\boldsymbol{\lambda}} \left\{ \log \left(\sum_{j=1}^{M} \sum_{t=1}^{T} L_j(t) \mathcal{N}(\boldsymbol{o}(t); \boldsymbol{M}_j \boldsymbol{\lambda}, \boldsymbol{\Sigma}_j) \right) \right\}$$

A more general scheme similar to eigenvoices that uses adaptive training is **cluster adaptive training**.

The idea of using a combination of basis models to represent a speaker space has also been applied to DNN adaptation in terms of a **weighted combination of multiple basis models**.

MAP Estimation

Maximum A-Posteriori (MAP) was extensively used for GMM-HMM speaker adaptation.

For model parameters by $\mathcal{M}$, MAP Estimation maximises

$$p(\mathcal{M}|\boldsymbol{O}) = \frac{p(\boldsymbol{O}|\mathcal{M})p_0(\mathcal{M})}{p(\boldsymbol{O})}$$

The influence of the *prior*, $p_0(\mathcal{M})$, may be explicitly seen. When a non-informative prior is used the MAP estimate becomes the ML estimate.

MAP estimation is very attractive as, given sufficient adaptation data, it will yield the same performance as a speaker-dependent system. The problem is the speed of adaptation.

The MAP estimate of the mean of state j is given by

$$\mu_{MAP} = \frac{\sigma_j^2 \mu_{pj} + \sigma_{pj}^2 \sum_{t=1}^{T} L_j(t) o(t)}{\sigma_j^2 + \sigma_{pj}^2 \sum_{t=1}^{T} L_j(t)}$$

where σ_j^2 is assumed known variance, μ_{pj} and σ_{pj}^2 are prior mean and prior variance of mean.

Could estimate μ_{pj} and σ_{pj}^2 from a set of speaker dependent models.

Alternatively the ratio $\frac{\sigma_j^2}{\sigma_{pj}^2} = \tau$ is set. Values of τ in the range 2-20 have been tried.

This only allows observed Gaussians to be updated. However, since there are high correlations between various states of the model sets, modifications to MAP have also been proposed.

Least Squares Linear Regression

Alternatively for GMM-HMMs use model adaptation based on a linear transform of the model parameters. Thus

$$\hat{\boldsymbol{\mu}}_j^{(s)} = \boldsymbol{A}^{(s)} \boldsymbol{\mu}_j + \boldsymbol{b}^{(s)}$$

where $\boldsymbol{A}^{(s)}$ is an $n \times n$ matrix and $\boldsymbol{b}^{(s)}$ is an $n \times 1$ vector. These describe the transform from the SI model to the particular speaker. This is sometimes written as

$$\hat{\boldsymbol{\mu}}_j^{(s)} = \boldsymbol{W}^{(s)} \boldsymbol{\xi}_j$$

where $\boldsymbol{W}^{(s)}$ is an $n \times (n+1)$ matrix and $\boldsymbol{\xi}_j$ is the extended mean vector

$$\boldsymbol{\xi} = \begin{bmatrix} 1 & \mu_{j1} & \ldots & \mu_{jn} \end{bmatrix}^T$$

Need to estimate a single (as above) transform per speaker or a small set of transformations for all Gaussians. This allows even unobserved Gaussians to be adapted.

One simple estimation scheme for $\boldsymbol{W}$ is *least squares*. The transform is found which minimises

$$\boldsymbol{W}^{(s)} = \arg\min_{\boldsymbol{W}} \left\{ \sum_{j=1}^{M} \sum_{t=1}^{T} L_j(t) \left(\boldsymbol{o}(t) - \boldsymbol{W}\boldsymbol{\xi}_j \right)' \left(\boldsymbol{o}(t) - \boldsymbol{W}\boldsymbol{\xi}_j \right) \right\}$$

This has a simple solution that

$$\boldsymbol{W}^{(s)} = \left(\sum_{j=1}^{M} \sum_{t=1}^{T} L_j(t) \boldsymbol{o}(t) \boldsymbol{\xi}_j' \right) \left(\sum_{j=1}^{M} \sum_{t=1}^{T} L_j(t) \boldsymbol{\xi}_j \boldsymbol{\xi}_j' \right)^{-1}$$

MLLR

The transformation matrix $\boldsymbol{W}^{(s)}$ may also be estimated such that the likelihood of the adaptation data is maximised (hence maximum likelihood linear regression). This is another example of EM training.

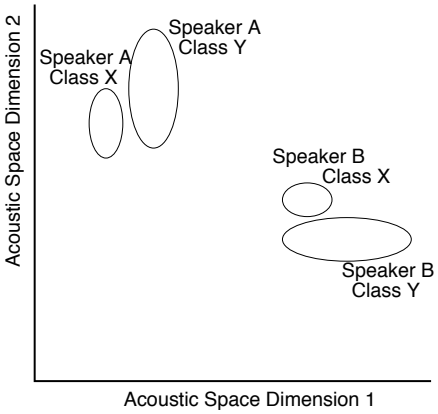
$$\boldsymbol{W}^{(s)} = \arg\max_{\boldsymbol{W}} \left\{ \sum_{j=1}^{M} \sum_{t=1}^{T} L_j(t) \log \left(\mathcal{N}(\boldsymbol{o}(t); \boldsymbol{W}\boldsymbol{\xi}_j, \boldsymbol{\Sigma}_j) \right) \right\}$$

Note: LSLR is identical to MLLR when $\boldsymbol{\Sigma}_j = \boldsymbol{I}$ for all states. MLLR has been found to outperform LSLR (and naturally fits into iterative EM schemes).

Rather than using a single transform multiple transforms may be used.

Assume that similar transforms relevant for Gaussians "close" in acoustic space

Can arrange Gaussians into a a **Regression Class Tree** and determine the number of transforms based on the amount of adaptation data available



The Transformation Matrix

The threshold used in the regression class tree will be a function of the number of parameters in the linear transformation.

1. **Simple Offset**: The transformation is

$$\hat{\boldsymbol{\mu}}_j^{(s)} = \boldsymbol{\mu}_j + \boldsymbol{b}^{(s)}$$

   Number of parameters is n.
2. **Diagonal**: Only the leading diagonal of the matrix and the bias are non-zero. Here all elements are independent of one another.

$$\hat{\mu}_{ji}^{(s)} = a_{ii}^{(s)} \mu_{ji} + b_i^{(s)}$$

   Number of parameters is $2n$.
3. **Full**: All elements of the matrix may be non-zero. Number of parameters is $n(n+1)$.
4. **Block Diagonal**: The matrix has a form like

$$\boldsymbol{A}^{(s)} = \begin{pmatrix} \boldsymbol{A}_s^{(s)} & \boldsymbol{0}^n & \boldsymbol{0}^n \\ \boldsymbol{0}^n & \boldsymbol{A}_\Delta^{(s)} & \boldsymbol{0}^n \\ \boldsymbol{0}^n & \boldsymbol{0}^n & \boldsymbol{A}_{\Delta^2}^{(s)} \end{pmatrix}$$

   The number of parameters, assuming 3 equal blocks and including the bias is $\left(\frac{n}{3}\right)^2 + n$

Block diagonal and full transformation matrices consistently outperform the diagonal case.

Form of the Adaptation Transform and CMLLR

Dominant approach for LVCSR GMM-HMMs: ML-based linear transformations. Several forms

- MLLR adaptation of the means discussed above

$$\hat{\mu}_m = \boldsymbol{A}\boldsymbol{\mu}_m + \boldsymbol{b}$$

- MLLR adaptation of the covariance matrices

$$\hat{\boldsymbol{\Sigma}}_m = \boldsymbol{H}\boldsymbol{\Sigma}_m\boldsymbol{H}'$$

- Constrained MLLR adaptation

$$\hat{\mu}_m = \boldsymbol{A}\boldsymbol{\mu}_m + \boldsymbol{b}; \quad \hat{\boldsymbol{\Sigma}}_m = \boldsymbol{A}\boldsymbol{\Sigma}_m\boldsymbol{A}'$$

In **Constrained MLLR**, the CMLLR Gaussian likelihood may be expressed as:

$$\mathcal{N}(\boldsymbol{o}; \boldsymbol{A}\boldsymbol{\mu}_m + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}_m\boldsymbol{A}') = \frac{1}{|\boldsymbol{A}|}\mathcal{N}(\boldsymbol{A}^{-1}\boldsymbol{o} - \boldsymbol{A}^{-1}\boldsymbol{b}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

This can also be generalised to using multiple transforms.

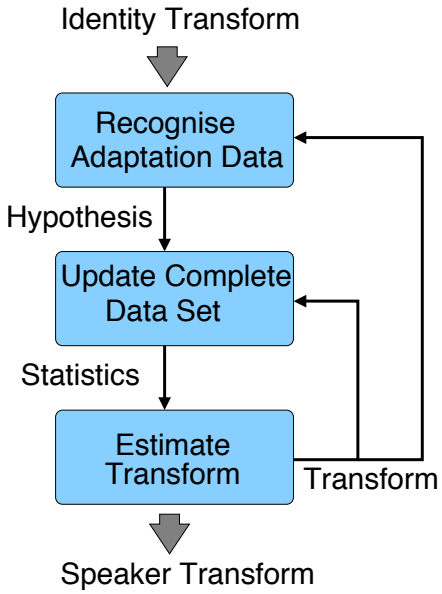
Note: a normalisation term is required when transforming features (though unnecessary for a *single* transform during recognition).

The CMLLR transform may be trained using ML via EM. The normalisation term complicates the optimisation, though a simple iterative scheme may be used.

Linear Transformation Estimation

Estimation of all the transforms is based on EM:

- requires the **transcription/hypothesis** of the adaptation data
- iterative process using "current" transform to estimate new transform



Two iterative loops for estimation:

1. estimate hypothesis given transform
2. update complete-dataset given transform and hypothesis

This is referred to as **Iterative MLLR**.

Can also use **lattice-based adaptation** rather than 1-best hypotheses (related to **confidence-score weighting** of words in 1-best)

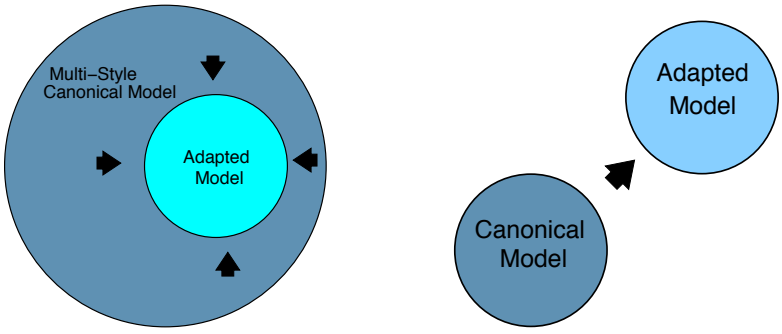
For supervised training hypothesis is known

- Can also vary complexity of transform with iteration

Training a "Good" Canonical Model

Standard multi-speaker (or multi-environment) canonical model

- treats all the data as a single "homogeneous" block
- model captures acoustic realisation of phones/words (desired)
- and acoustic environment, speaker, speaking style variations (unwanted)

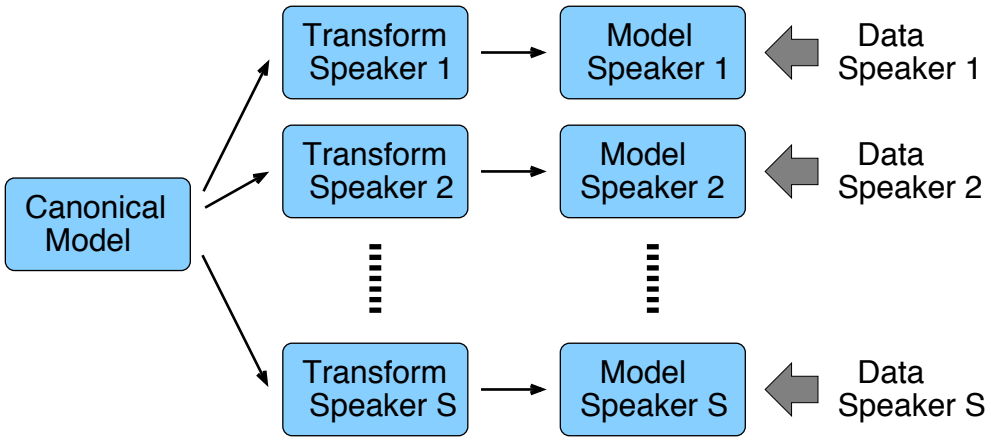


Multi-Speaker/Environment Adaptive

Two different forms of canonical model:

- **Multi-Speaker/Environment**: adaptation converts a general system to a specific condition;
- **Adaptive**: adaptation converts a "neutral" system to a specific condition

Adaptive Training



In adaptive training the training corpus is split into "homogeneous" blocks

- use adaptation transforms to represent unwanted acoustic factors
- canonical model only represents desired variability

All forms of linear transform can be used for adaptive training for GMM-HMMs

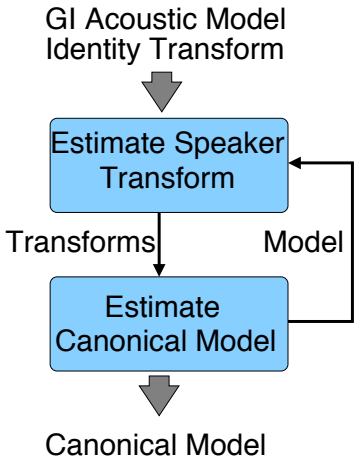
- CMLLR adaptive training highly efficient
  (if single transform, can modify features & apply more broadly)

CMLLR Adaptive Training

- The CMLLR likelihood for a Gaussian may be expressed as:

$$\mathcal{N}(\boldsymbol{o}; \boldsymbol{A}\boldsymbol{\mu}_m + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}_m\boldsymbol{A}') = \frac{1}{|\boldsymbol{A}|}\mathcal{N}(\boldsymbol{A}^{-1}\boldsymbol{o} - \boldsymbol{A}^{-1}\boldsymbol{b}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

same as feature normalisation - simply train model in transformed space



Interleave Model and transform estimation This process is:

- estimate model given transforms
- estimate transform given model and input transform

Note adaptive canonical HMM-GMM model not suited for unadapted initial decode

- Speaker independent model used for initial hypothesis

DNN Adaptation

Three Main Approaches

1. Normalise input features
   - VTLN
   - estimate CMLLR feature transforms (using GMM system ...)
2. Adapt DNN model parameters / change structure
   - add speaker-specific layer and train with EBP: **Linear Input layer or LIN**
   - **multi-basis structured network** adaptively weights sub-networks for different speakers — similar idea to eigenvoices
   - adaptive activation functions **Learning Hidden Unit Contributions or LHUC** & **Parameterised Sigmoid / ReLU**
   - adapt weights using **KL-Divergence regularisation**.
     Controls KLD of outputs between original/adapted weights.
     Same as interpolating target distribution with distribution from SI model
3. DNN auxiliary inputs
   - train/test with an extra vector that describes the speaker/environment
   - *i*-vector approach (from speaker recognition). Captures main variations in a speaker space (30-100 dim) like eigenvoices. No supervision hypothesis required

All of these methods are used (and sometimes in combination)

Generally improvements from adaptation are less than with GMM-HMM systems.

Parameterised Sigmoid/ReLU DNN Activation Functions

General parameterised Sigmoid/ReLU (or *p*-Sigmoid/ReLU) applies additional scale factor to input (& bias) to activation function and also scales output.

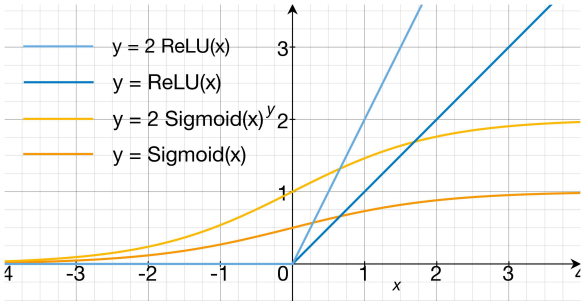
These additional activation function parameters can be made speaker specific.

If just use activation function scaling

- *p*-Sigmoid/*p*-ReLU function adaptation of DNN node *i* uses

$$f_i^s(a_i) = \alpha_i^s \cdot f(a_i)$$

  where α_i^s is linear scale for speaker *s*.
- Similar to LHUC adaptation framework



- More hidden layers are progressively used for adaptation (from input layers first)
- Found that adapting more hidden layers consistently reduces WER.
- DNN layers closer to input model low level characteristics of speech and are more important in speaker adaptation.

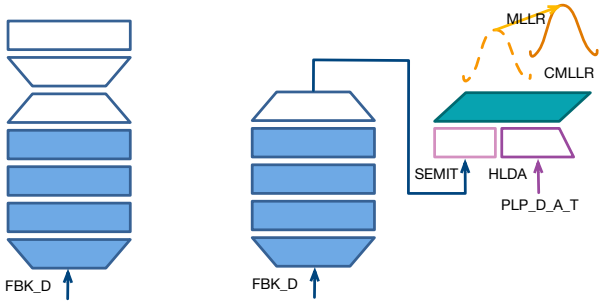
Combined Methods of DNN Adaption

Many combination approaches have been applied to DNN adaptation e.g.

- CMLLR feature adaptation and i-vectors
- multi-basis structured network and i-vectors
- ...

Further combined adaptation possible if using DNN bottleneck layer features in GMMs:

- Train bottleneck layer DNN to estimate GMM features as usual.
- BN GMM-HMMs trained can apply CMLLR-based SAT
- Can apply layer-by-layer adaptation using *p*-sigmoid or *p*-ReLU to bottleneck feature extraction
- Can also apply MLLR test adaptation!



Overall there are very many ways of combining the adaptation approaches for different types of models!

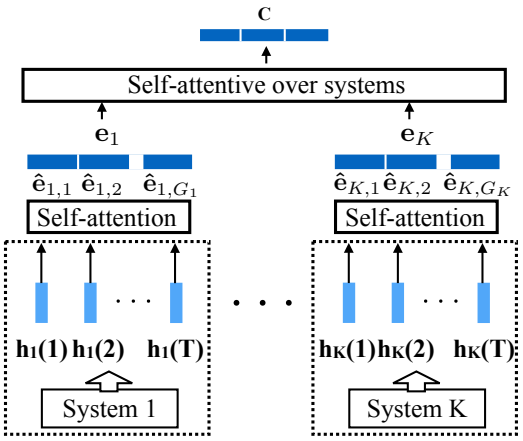
DNN-based Speaker Embeddings

The i-vector (& eigenvoice/CAT) approaches yield a **speaker space**.

- Points in that space can be thought of as a **speaker embedding**.
- Can be used in **speaker recognition** and **speaker diarisation**.
- Speaker embedding can be explicitly estimated using a DNN and a bottleneck layer (known as a **d-vector**)

This d-vector extraction works by

- Train speaker discrimination DNN (predict frame speaker label from input) with bottleneck layer (d-vector)
- Pool or use **self-attention** to get segment level d-vector
- Training ensures that cosine-distance is appropriate (for clustering in diaristaion)
- Can also combine multiple multi-head attention models with further self-attention (see fig)



Speaker Diarisation

Diarisation finds **who spoke when** in a stream of multi-speaker audio.
Normally performed in a sequence of stages:

- Speech/non-speech detection (i.e. VAD)
- **Change-point** detection
- **Clustering of segments** to find groups that should correspond to individual speakers

Recent diarisation systems uses **speaker embeddings** (d-vectors etc) to represent segments

- extracted from overlapped fixed length windows of 1-2s (segment boundaries unknown)
- embeddings used to find change points (can feed into a separate DNN model) alternative for change points is finding KL divergence between sliding Gaussian windows

Many **clustering** approaches to group speakers

- normally unsupervised (& unknown number of speakers) e.g. **spectral clustering**
- recent interest in supervised clustering e.g. **discriminative neural clustering**

A key issue is **speaker overlap** which makes all of the stages harder.

- can be helped if have multiple microphones / beamformer information

Diarisation is used as a front-end for speech recognition systems, both for identifying regions for adaptation and identifying when the same speaker occurs in a stream of audio, cf. **speaker attributed speech recognition**.

Summary

- Adaptation and normalisation can improve speech recognition systems
- Can be used in an unsupervised fashion
- Clustering, MAP-based and Linear Transform approaches (and combinations) all possible
- CMLLR with single transforms can be applied to features
- Adaptive training aims to produce a more "compact" canonical model
- CMLLR adaptive training - efficiently handles non-homogeneous data

Generally

- simple ASR systems - larger reductions in WER from adaptation
- more front-end normalisation gives smaller WER reductions from adaptation
- greater training/test mismatch: larger WER reductions

DNN adaptation approaches:

- use feature normalisation, modify network parameters, add extra layers, add parallel sub-networks with speaker-specific weights, use learnable speaker-specific parameterised activation functions or add extra network inputs, and combinations of all of these!

Speaker embeddings allow the generation of a speaker space that is used in speaker recognition and diarisation.

- components can be directly useful in speech recognition systems

While approaches have been described in terms of speaker adaptation, can handle **other types of mismatch** between training & test (e.g. acoustic environment)

References

- T. Anastasakos, J. McDonough, R. Schwartz, & J. Makhoul. "A Compact Model for Speaker-Adaptive Training." *Proc. ICSLP*, 1996.
- M.J.F. Gales. "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition." *Computer Speech & Language*, vol 12, pp. 75–98, 1998.
- J-L. Gauvain and C-H. Lee. "Maximum *a-posteriori* Estimation for Multivariate Gaussian Mixture Observations of Markov Chains." *IEEE Transactions SAP*, Vol. 2, pp. 291–298, 1994.
- Q. Li, F.L. Kreyssig, C. Zhang & P.C. Woodland. "Discriminative Neural Clustering for Speaker Diarisation". Proc. IEEE SLT Workshop, 2021.
- P. Karanasou, C. Wu, M. Gales & P. Woodland. "I-Vectors and Structured Neural Networks for Rapid Adaptation of Acoustic Models". *IEEE/ACM IEEE/ACM Transactions ASLP*, Vol. 25. No. 4, 2017
- R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. "Eigenvoices for Speaker Adaptation." *Proc. ICSLP*, 1998.
- C.J. Leggetter & P.C. Woodland. "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs." *Computer Speech & Language*, vol. 9, pp. 171–186, 1995.
- G. Saon, H. Soltau, D. Nahamoo, & M. Picheny. "Speaker adaptation of Neural Network Acoustic Models using i-Vectors." *Proc. IEEE ASRU*, 2013.
- G. Sun, C. Zhang, P. Woodland. "Combination of Deep Speaker Embeddings for Diarisation". arXiv:2010.12025, 2021.
- P. Swietojanski, J. Li, & S. Renals. "Learning Hidden Unit Contributions for Unsupervised Acoustic Model Adaptation." *IEEE/ACM Transactions ASLP*, Vol. 24, No. 8, 2016.
- P.C. Woodland. "Speaker Adaptation for Continuous Density HMMs: A Review." *Proc. ISCA ITRW on Adaptation Methods for Speech Recognition*, 2001.
- D. Yu, K. Yao, H. Su, G. Li & F. Seide. "KL-Divergence Regularized Neural Network Adaptation for Improved Large Vocabulary Recognition." *Proc. ICASSP*, 2013.
- C. Zhang & P.C. Woodland. "DNN Speaker Adaptation using Parameterised Sigmoid and ReLU Hidden Activation Functions." *Proc. ICASSP*, 2016.