Speaker Adaptation

Steve Renals

Automatic Speech Recognition – ASR Lectures 13&14
5, 9 March 2015

Overview

Speaker Adaptation

- Introduction: speaker-specific variation, modes of adaptation
- Model-based adaptation: MAP
- Model-based adaptation: MLLR
- Model-based adaptation: Speaker space models
- Speaker normalization: VTLN
- Adaptive training
- Adaptation for hybrid HMM / NN systems

Speaker independent / dependent / adaptive

- Speaker independent (SI) systems have long been the focus for research in transcription, dialogue systems, etc.
- Speaker dependent (SD) systems can result in word error rates 2–3 times lower than SI systems (given the same amount of training data)
- A Speaker adaptive (SA) system... we would like
  - Error rates similar to SD systems
  - Building on an SI system
  - Requiring only a small fraction of the speaker-specific training data used by an SD system

Speaker-specific variation

- Acoustic model
  - Speaking styles
  - Accents
  - Speech production anatomy (eg length of the vocal tract)

  Also non-speaker variation, such as channel conditions (telephone, reverberant room, close talking mic) and application domain

  Speaker adaptation of acoustic models aims to reduce the mismatch between test data and the models

Speaker-specific variation

- Acoustic model
  - Speaking styles
  - Accents
  - Speech production anatomy (eg length of the vocal tract)

  Also non-speaker variation, such as channel conditions (telephone, reverberant room, close talking mic) and application domain

  Speaker adaptation of acoustic models aims to reduce the mismatch between test data and the models
- Pronunciation model: speaker-specific, consistent change in pronunciation

Speaker-specific variation

- Acoustic model
  - Speaking styles
  - Accents
  - Speech production anatomy (eg length of the vocal tract)

  Also non-speaker variation, such as channel conditions (telephone, reverberant room, close talking mic) and application domain

  Speaker adaptation of acoustic models aims to reduce the mismatch between test data and the models
- Pronunciation model: speaker-specific, consistent change in pronunciation
- Language model: user-specific documents (exploited in personal dictation systems)

Modes of adaptation

- Supervised or unsupervised
  - Supervised: the word level transcription of the adaptation data is known (and HMMs may be constructed)
  - Unsupervised: the transcription must be estimated (eg using recognition output)

Modes of adaptation

- Supervised or unsupervised
  - Supervised: the word level transcription of the adaptation data is known (and HMMs may be constructed)
  - Unsupervised: the transcription must be estimated (eg using recognition output)
- Static or dynamic
  - Static: All adaptation data is presented to the system in a block before the final system is estimated (eg as used in enrollment in a dictation system)
  - Dynamic: Adaptation data is incrementally available, and models must be adapted before all adaptation data is available (eg as used in a spoken dialogue system)

Approaches to adaptation

- Model based: Adapt the parameters of the acoustic models to better match the observed data
  - Maximum a posteriori (MAP) adaptation of HMM/GMM parameters
  - Maximum likelihood linear regression (MLLR) of Gaussian parameters
  - Linear input network (LIN) for neural networks

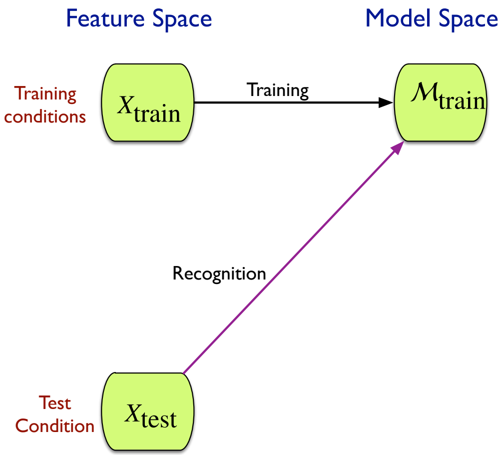
Approaches to adaptation

- Model based: Adapt the parameters of the acoustic models to better match the observed data
  - Maximum a posteriori (MAP) adaptation of HMM/GMM parameters
  - Maximum likelihood linear regression (MLLR) of Gaussian parameters
  - Linear input network (LIN) for neural networks
- Speaker Normalization: Normalize the acoustic data to reduce mismatch with the acoustic models
  - Vocal Tract Length Normalization (VTLN)
  - Constrained MLLR (cMLLR) — model-based normalisation!

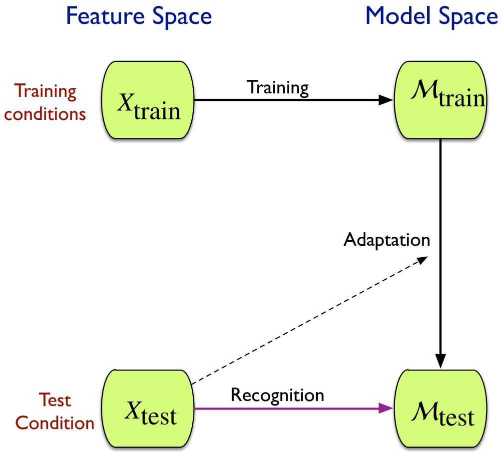
Approaches to adaptation

- Model based: Adapt the parameters of the acoustic models to better match the observed data
  - Maximum a posteriori (MAP) adaptation of HMM/GMM parameters
  - Maximum likelihood linear regression (MLLR) of Gaussian parameters
  - Linear input network (LIN) for neural networks
- Speaker Normalization: Normalize the acoustic data to reduce mismatch with the acoustic models
  - Vocal Tract Length Normalization (VTLN)
  - Constrained MLLR (cMLLR) — model-based normalisation!
- Speaker space: Estimate multiple sets of acoustic models, characterizing new speakers in terms of these model sets
  - Cluster-adapative training
  - Eigenvoices
  - Speaker codes

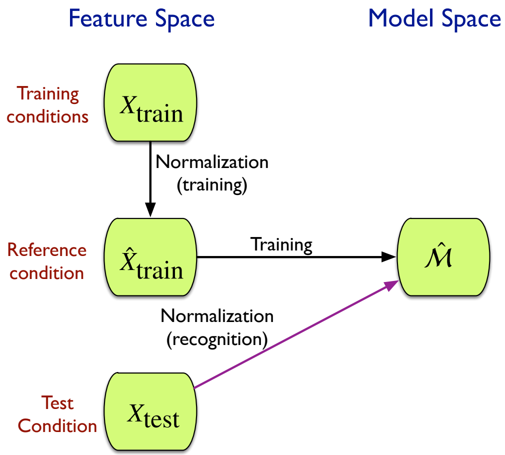
Adaptation and normalization of acoustic models



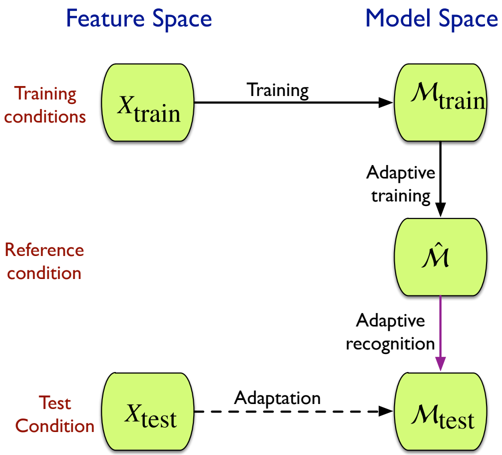
Adaptation and normalization of acoustic models



Adaptation and normalization of acoustic models



Adaptation and normalization of acoustic models



Desirable properties for model-based speaker adaptation

- **Compact:** relatively few speaker-dependent parameters
- **Unsupervised:** does not require labelled adaptation data, or changes to the training
- **Efficient:** low computational requirements
- **Flexible:** applicable to different model variants

Model-based adaptation: The MAP family

- Basic idea Use the SI models as a prior probability distribution over model parameters when estimating using speaker-specific data

Model-based adaptation: The MAP family

- Basic idea Use the SI models as a prior probability distribution over model parameters when estimating using speaker-specific data
- Theoretically well-motivated approach to incorporating the knowledge inherent in the SI model parameters

Model-based adaptation: The MAP family

- Basic idea Use the SI models as a prior probability distribution over model parameters when estimating using speaker-specific data
- Theoretically well-motivated approach to incorporating the knowledge inherent in the SI model parameters
- Maximum likelihood (ML) training sets the model parameters $\boldsymbol{\lambda}$ to maximize the likelihood $p(\mathbf{X} \mid \boldsymbol{\lambda})$

Model-based adaptation: The MAP family

- Basic idea Use the SI models as a prior probability distribution over model parameters when estimating using speaker-specific data
- Theoretically well-motivated approach to incorporating the knowledge inherent in the SI model parameters
- Maximum likelihood (ML) training sets the model parameters $\boldsymbol{\lambda}$ to maximize the likelihood $p(\mathbf{X} \mid \boldsymbol{\lambda})$
- Maximum a posteriori (MAP) training maximizes the posterior of the parameters given the data:

$$p(\boldsymbol{\lambda} \mid \mathbf{X}) \propto p(\mathbf{X} \mid \boldsymbol{\lambda}) p_0(\boldsymbol{\lambda})$$

$p_0(\boldsymbol{\lambda})$ is the prior distribution of the parameters

Model-based adaptation: The MAP family

- Basic idea Use the SI models as a prior probability distribution over model parameters when estimating using speaker-specific data
- Theoretically well-motivated approach to incorporating the knowledge inherent in the SI model parameters
- Maximum likelihood (ML) training sets the model parameters $\boldsymbol{\lambda}$ to maximize the likelihood $p(\mathbf{X} \mid \boldsymbol{\lambda})$
- Maximum a posteriori (MAP) training maximizes the posterior of the parameters given the data:

$$p(\boldsymbol{\lambda} \mid \mathbf{X}) \propto p(\mathbf{X} \mid \boldsymbol{\lambda}) p_0(\boldsymbol{\lambda})$$

  $p_0(\boldsymbol{\lambda})$ is the prior distribution of the parameters
- The use of a prior distribution, based on the SI models, means that less data is required to estimate the speaker-specific models: we are not starting from complete ignorance

Recall: ML estimation of GMM/HMM

- The mean of the mth Gaussian component of the jth state is estimated using a weighted average

$$\boldsymbol{\mu}_{mj} = \frac{\sum_n \gamma_{jm}(n)\mathbf{x}_n}{\sum_n \gamma_{jm}(n)}$$

- Where $\sum_n \gamma_{jm}(n)$ is the component occupation probability
- The covariance of the Gaussian component is given by:

$$\boldsymbol{\Sigma}_{mj} = \frac{\sum_n \gamma_{jm}(n)(\mathbf{x}_n - \boldsymbol{\mu}_{jm})(\mathbf{x}_n - \boldsymbol{\mu}_{jm})^T}{\sum_n \gamma_{jm}(n)}$$

MAP estimation

- What is $p_0(\boldsymbol{\lambda})$?

MAP estimation

- What is $p_0(\boldsymbol{\lambda})$?
- Conjugate prior: the prior distribution has the same form as the posterior. There is no simple conjugate prior for GMMs, but an intuitively understandable approach may be employed.

MAP estimation

- What is $p_0(\boldsymbol{\lambda})$?
- Conjugate prior: the prior distribution has the same form as the posterior. There is no simple conjugate prior for GMMs, but an intuitively understandable approach may be employed.
- If the prior mean is $\boldsymbol{\mu}_0$, then the MAP estimate for the adapted mean $\hat{\boldsymbol{\mu}}$ of Gaussian is given by:

$$\hat{\boldsymbol{\mu}} = \frac{\tau \boldsymbol{\mu}_0 + \sum_n \gamma(n)\mathbf{x}_n}{\tau + \sum_n \gamma(n)}$$

MAP estimation

- What is $p_0(\boldsymbol{\lambda})$?
- Conjugate prior: the prior distribution has the same form as the posterior. There is no simple conjugate prior for GMMs, but an intuitively understandable approach may be employed.
- If the prior mean is $\boldsymbol{\mu}_0$, then the MAP estimate for the adapted mean $\hat{\boldsymbol{\mu}}$ of Gaussian is given by:

$$\hat{\boldsymbol{\mu}} = \frac{\tau \boldsymbol{\mu}_0 + \sum_n \gamma(n)\mathbf{x}_n}{\tau + \sum_n \gamma(n)}$$

  - τ is a *hyperparameter* that controls the balance between the ML estimate of the mean, its prior value. Typically τ is in the range 2–20

MAP estimation

- What is $p_0(\boldsymbol{\lambda})$?
- Conjugate prior: the prior distribution has the same form as the posterior. There is no simple conjugate prior for GMMs, but an intuitively understandable approach may be employed.
- If the prior mean is $\boldsymbol{\mu}_0$, then the MAP estimate for the adapted mean $\hat{\boldsymbol{\mu}}$ of Gaussian is given by:

$$\hat{\boldsymbol{\mu}} = \frac{\tau \boldsymbol{\mu}_0 + \sum_n \gamma(n) \mathbf{x}_n}{\tau + \sum_n \gamma(n)}$$

  - τ is a *hyperparameter* that controls the balance between the ML estimate of the mean, its prior value. Typically τ is in the range 2–20
  - $\mathbf{x}_n$ is the adaptation vector at time n

MAP estimation

- What is $p_0(\boldsymbol{\lambda})$?
- Conjugate prior: the prior distribution has the same form as the posterior. There is no simple conjugate prior for GMMs, but an intuitively understandable approach may be employed.
- If the prior mean is $\boldsymbol{\mu}_0$, then the MAP estimate for the adapted mean $\hat{\boldsymbol{\mu}}$ of Gaussian is given by:

$$\hat{\boldsymbol{\mu}} = \frac{\tau \boldsymbol{\mu}_0 + \sum_n \gamma(n) \mathbf{x}_n}{\tau + \sum_n \gamma(n)}$$

  - τ is a *hyperparameter* that controls the balance between the ML estimate of the mean, its prior value. Typically τ is in the range 2–20
  - $\mathbf{x}_n$ is the adaptation vector at time n
  - $\gamma(n)$ the probability of this Gaussian at this time

MAP estimation

- What is $p_0(\boldsymbol{\lambda})$?
- Conjugate prior: the prior distribution has the same form as the posterior. There is no simple conjugate prior for GMMs, but an intuitively understandable approach may be employed.
- If the prior mean is $\boldsymbol{\mu}_0$, then the MAP estimate for the adapted mean $\hat{\boldsymbol{\mu}}$ of Gaussian is given by:

$$\hat{\boldsymbol{\mu}} = \frac{\tau \boldsymbol{\mu}_0 + \sum_n \gamma(n) \mathbf{x}_n}{\tau + \sum_n \gamma(n)}$$

  - τ is a *hyperparameter* that controls the balance between the ML estimate of the mean, its prior value. Typically τ is in the range 2–20
  - $\mathbf{x}_n$ is the adaptation vector at time n
  - $\gamma(n)$ the probability of this Gaussian at this time
- As the amount of training data increases, so the MAP estimate converges to the ML estimate

Local estimation

- Basic idea The main drawback to MAP adaptation is that it is local

Local estimation

- Basic idea The main drawback to MAP adaptation is that it is local
- Only the parameters belonging to Gaussians of observed states will be adapted

Local estimation

- Basic idea The main drawback to MAP adaptation is that it is local
- Only the parameters belonging to Gaussians of observed states will be adapted
- Large vocabulary speech recognition systems have about 10^5 Gaussians: most will not be adapted

Local estimation

- Basic idea The main drawback to MAP adaptation is that it is local
- Only the parameters belonging to Gaussians of observed states will be adapted
- Large vocabulary speech recognition systems have about 10^5 Gaussians: most will not be adapted
  - Structural MAP (SMAP) approaches have been introduced to share Gaussians

Local estimation

- Basic idea The main drawback to MAP adaptation is that it is local
- Only the parameters belonging to Gaussians of observed states will be adapted
- Large vocabulary speech recognition systems have about 10^5 Gaussians: most will not be adapted
  - Structural MAP (SMAP) approaches have been introduced to share Gaussians
  - The MLLR family of adaptation approaches addresses this by assuming that transformations for a specific speaker are systematic across Gaussians, states and models

Local estimation

- Basic idea The main drawback to MAP adaptation is that it is local
- Only the parameters belonging to Gaussians of observed states will be adapted
- Large vocabulary speech recognition systems have about 10^5 Gaussians: most will not be adapted
  - Structural MAP (SMAP) approaches have been introduced to share Gaussians
  - The MLLR family of adaptation approaches addresses this by assuming that transformations for a specific speaker are systematic across Gaussians, states and models
- MAP adaptation is very useful for domain adaptation:

Local estimation

- Basic idea The main drawback to MAP adaptation is that it is local
- Only the parameters belonging to Gaussians of observed states will be adapted
- Large vocabulary speech recognition systems have about 10^5 Gaussians: most will not be adapted
  - Structural MAP (SMAP) approaches have been introduced to share Gaussians
  - The MLLR family of adaptation approaches addresses this by assuming that transformations for a specific speaker are systematic across Gaussians, states and models
- MAP adaptation is very useful for domain adaptation:
  - Example: adapting a conversational telephone speech system (100s of hours of data) to multiparty meetings (10s of hours of data) works well with MAP

SMAP: Structural MAP

- Basic idea share Gaussians by organising them in a tree, whose root contains all the Gaussians
- At each node in the tree compute mean offset and diagonal variance scaling term
- For each node, its parent is used as a prior distribution
- This has been shown to speed adaptation compared with standard MAP, while converging to the same solution as standard MAP in the large data limit

The Linear Transform family

- Basic idea Rather than directly adapting the model parameters, estimate a transform which may be applied the Gaussian means and covariances

The Linear Transform family

- Basic idea Rather than directly adapting the model parameters, estimate a transform which may be applied the Gaussian means and covariances
- Linear transform applied to parameters of a set of Gaussians: adaptation transform parameters are shared across Gaussians

The Linear Transform family

- Basic idea Rather than directly adapting the model parameters, estimate a transform which may be applied the Gaussian means and covariances
- Linear transform applied to parameters of a set of Gaussians: adaptation transform parameters are shared across Gaussians
- This addresses the locality problem arising in MAP adaptation, since each adaptation data point can affect many of (or even all) the Gaussians in the system

The Linear Transform family

- Basic idea Rather than directly adapting the model parameters, estimate a transform which may be applied the Gaussian means and covariances
- Linear transform applied to parameters of a set of Gaussians: adaptation transform parameters are shared across Gaussians
- This addresses the locality problem arising in MAP adaptation, since each adaptation data point can affect many of (or even all) the Gaussians in the system
- There are relatively few adaptation parameters, so estimation is robust

The Linear Transform family

- Basic idea Rather than directly adapting the model parameters, estimate a transform which may be applied the Gaussian means and covariances
- Linear transform applied to parameters of a set of Gaussians: adaptation transform parameters are shared across Gaussians
- This addresses the locality problem arising in MAP adaptation, since each adaptation data point can affect many of (or even all) the Gaussians in the system
- There are relatively few adaptation parameters, so estimation is robust
- Maximum Likelihood Linear Regression (MLLR) is the best known linear transform approach to speaker adaptation

MLLR: Maximum Likelihood Linear Regression

- MLLR is the best known linear transform approach to speaker adaptation
- Affine transform of mean parameters

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

  If the observation vectors are d-dimension, then $\boldsymbol{A}$ is a $d \times d$ matrix and $\mathbf{b}$ is d-dimension vector

- If we define $\mathbf{W} = [\mathbf{bA}]$ and $\boldsymbol{\eta} = [1\boldsymbol{\mu}^T]^T$, then we can write:

$$\hat{\boldsymbol{\mu}} = \mathbf{W}\boldsymbol{\eta}$$

- In MLLR, $\mathbf{W}$ is estimated so as to maximize the likelihood of the adaptation data
- A single transform $\mathbf{W}$ can be shared across a set of Gaussian components (even all of them!)

Regression classes

- The number of transforms may obtained automatically
- A set of Gaussian components that share a transform is called a regression class
- Obtain the regression classes by constructing a *regression class tree*
- Each node in the tree represents a regression class sharing a transform
- For an adaptation set, work down the tree until arriving at the most specific set of nodes for which there is sufficient data
- Regression class tree constructed in a similar way to state clustering tree
- In practice the number of regression may be very small: one per context-independent phone class, one per broad class, or even just two (speech/non-speech)

Estimating the transforms

- The linear transformation matrix W is obtained by finding its setting which optimizes the log likelihood
- Mean adaptation: Log likelihood

$$L = \sum_r \sum_n \gamma_r(n) \log \left(K_r \exp \left(-\frac{1}{2} (\mathbf{x}_n - \mathbf{W}\boldsymbol{\eta}_r)^T \boldsymbol{\Sigma}_r^{-1} (\mathbf{x}_n - \mathbf{W}\boldsymbol{\eta}_r) \right) \right)$$

  where r ranges over the components belonging to the regression class
- Differentiating L and setting to 0 results in an equation for $\mathbf{W}$: there is no closed form solution if $\boldsymbol{\Sigma}$ is full covariance; can be solved if $\boldsymbol{\Sigma}$ is diagonal (but requires a matrix inversion)
- Variance adaptation is also possible
- See Gales and Woodland (1996), Gales (1998) for details

MLLR in practice

- Mean-only MLLR results in 10–15% relative reduction in WER
- Few regression classes and well-estimated transforms work best in practice
- Robust adaptation available with about 1 minute of speech; performance similar to SD models available with 30 minutes of adaptation data
- Such linear transforms can account for any systematic (linear) variation from the speaker independent models, for example those caused by channel effects.

Constrained MLLR (cMLLR)

- Basic idea use the same linear transform for both mean and covariance

$$\hat{\mu} = \mathbf{A}'\mu - \mathbf{b}'$$
$$\hat{\mathbf{\Sigma}} = \mathbf{A}'\mathbf{\Sigma}\mathbf{A}'^T$$

Constrained MLLR (cMLLR)

- Basic idea use the same linear transform for both mean and covariance

$$\hat{\mu} = \mathbf{A}'\mu - \mathbf{b}'$$
$$\hat{\mathbf{\Sigma}} = \mathbf{A}'\mathbf{\Sigma}\mathbf{A}'^{T}$$

- No closed form solution but can be solved iteratively

Constrained MLLR (cMLLR)

- Basic idea use the same linear transform for both mean and covariance

$$\hat{\mu} = \mathbf{A}'\mu - \mathbf{b}'$$
$$\hat{\mathbf{\Sigma}} = \mathbf{A}'\mathbf{\Sigma}\mathbf{A}'^T$$

- No closed form solution but can be solved iteratively
- Log likelihood for cMLLR

$$L = \mathcal{N}(\mathbf{A}\mathbf{x}_n + \mathbf{b}; \mu, \mathbf{\Sigma}) + \log(|\mathbf{A}|) \qquad \mathbf{A}' = \mathbf{A}^{-1}; \mathbf{b}' = \mathbf{A}\mathbf{b}$$

Equivalent to applying the linear transform to the data!
Also called fMLLR (feature space MLLR)

Constrained MLLR (cMLLR)

- Basic idea use the same linear transform for both mean and covariance

$$\hat{\mu} = \mathbf{A}'\mu - \mathbf{b}'$$
$$\hat{\mathbf{\Sigma}} = \mathbf{A}'\mathbf{\Sigma}\mathbf{A}'^{T}$$

- No closed form solution but can be solved iteratively
- Log likelihood for cMLLR

$$L = \mathcal{N}(\mathbf{A}\mathbf{x}_n + \mathbf{b}; \mu, \mathbf{\Sigma}) + \log(|\mathbf{A}|) \qquad \mathbf{A}' = \mathbf{A}^{-1}; \mathbf{b}' = \mathbf{A}\mathbf{b}$$

  Equivalent to applying the linear transform to the data!
  Also called fMLLR (feature space MLLR)
- Iterative solution amenable to online/dynamic adaptation, by using just one iteration for each increment

Constrained MLLR (cMLLR)

- Basic idea use the same linear transform for both mean and covariance

$$\hat{\mu} = \mathbf{A}'\mu - \mathbf{b}'$$
$$\hat{\mathbf{\Sigma}} = \mathbf{A}'\mathbf{\Sigma}\mathbf{A}'^{T}$$

- No closed form solution but can be solved iteratively
- Log likelihood for cMLLR

$$L = \mathcal{N}(\mathbf{A}\mathbf{x}_n + \mathbf{b}; \mu, \mathbf{\Sigma}) + \log(|\mathbf{A}|) \qquad \mathbf{A}' = \mathbf{A}^{-1}\,;\ \mathbf{b}' = \mathbf{A}\mathbf{b}$$

  Equivalent to applying the linear transform to the data!
  Also called fMLLR (feature space MLLR)

- Iterative solution amenable to online/dynamic adaptation, by using just one iteration for each increment
- Similar improvement in accuracy to standard MLLR

Speaker-adaptive training (SAT)

- Basic idea Rather than SI seed (canonical) models, construct models designed for adaptation
- Estimate parameters of canonical models by training MLLR mean transforms for each training speaker
- Train using the MLLR transform for each speaker; interleave Gaussian parameter estimation and MLLR transform estimation
- SAT results in much higher training likelihoods, and improved recognition results
- But: increased training complexity and storage requirements
- SAT using cMLLR, corresponds to a type of speaker normalization at training time

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers
- Speaker clustering: sets of HMMs for different speaker clusters

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers
- Speaker clustering: sets of HMMs for different speaker clusters
- Drawbacks:

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers
- Speaker clustering: sets of HMMs for different speaker clusters
- Drawbacks:
  - Hard division of speakers into groups

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers
- Speaker clustering: sets of HMMs for different speaker clusters
- Drawbacks:
  - Hard division of speakers into groups
  - Fragments training data

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers
- Speaker clustering: sets of HMMs for different speaker clusters
- Drawbacks:
  - Hard division of speakers into groups
  - Fragments training data
- Weighted speaker cluster approaches which use an interpolated model to represent the current speaker

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers
- Speaker clustering: sets of HMMs for different speaker clusters
- Drawbacks:
  - Hard division of speakers into groups
  - Fragments training data
- Weighted speaker cluster approaches which use an interpolated model to represent the current speaker
  - Cluster-adaptive training

Speaker Space Methods

- Gender-dependent models: sets of HMMs for male and for female speakers
- Speaker clustering: sets of HMMs for different speaker clusters
- Drawbacks:
  - Hard division of speakers into groups
  - Fragments training data
- Weighted speaker cluster approaches which use an interpolated model to represent the current speaker
  - Cluster-adaptive training
  - Eigenvoices

Cluster-adaptive training

- Basic idea Represent a speaker as a weighted sum of speaker cluster models
- Different cluster models have shared variances and mixture weights, but separate means
- For a new speaker, mean is defined as

$$\boldsymbol{\mu} = \sum_c \lambda_c \boldsymbol{\mu}_c$$

- Given the canonical models, only the λ_c mixing parameters need estimated for each speaker
- Given sets of weights for individual speakers, means of the clusters may be updated
- CAT can reduce WER in large vocabulary tasks by about 4–8% relative
- For more, see Gales (2000), Cluster adaptive training of hidden Markov models, *IEEE Trans Speech and Audio Processing*, **8**:417–428.

Eigenvoices

- Basic idea Construct a speaker space from a set of SD HMMs
- Could regard each canonical model as forming a dimension of speaker space
- Generalize by computing PCA of sets of "supervectors" (concatenated mean vectors), to form speaker space: each dimension is an "eigenvoice"
- Represent a new speaker as a combination of eigenvoices
- Close relation to CAT
- Computationally intensive, does not scale well to large vocabulary systems
- For more, see Kuhn et al (2000), Rapid speaker adaptation in eigenvoice space, *IEEE Trans Speech and Audio Processing*, **8**:695–707.

Feature normalization

- Basic idea: Transform the features to reduce mismatch between training and test
- *Cepstral Mean Normalization* (CMN): subtract the avergae feature value from each feature, so each feature has a mean value of 0. makes features robust to some linear filtering of the signal (channel variation)
- *Cepstral Variance Normalization* (CVN): Divide feature vector by standard deviation of feature vectors, so each feature vector element has a variance of 1
- Cepstral mean and variance normalisation, CMN/CVN:

$$\hat{\mathbf{x}}_i = \frac{\mathbf{x}_i - \boldsymbol{\mu}(\mathbf{x})}{\boldsymbol{\sigma}(\mathbf{x})}$$

- Compute mean and variance statistics over longest available segments with the same speaker/channel
- Real time normalisation: compute a moving average

Vocal Tract Length Normalization (VTLN)

- Basic idea Normalize the acoustic data to take account of changes in vocal tract length

Vocal Tract Length Normalization (VTLN)

- Basic idea Normalize the acoustic data to take account of changes in vocal tract length
- Vocal tract length (VTL):
  - First larynx descent in first 2-3 years of life
  - VTL grows according to body size, and is sex-dependent
  - Puberty: second larynx descent for males

Vocal Tract Length Normalization (VTLN)

- Basic idea Normalize the acoustic data to take account of changes in vocal tract length
- Vocal tract length (VTL):
  - First larynx descent in first 2-3 years of life
  - VTL grows according to body size, and is sex-dependent
  - Puberty: second larynx descent for males
- VTL has large effect on the spectrum
  - Tube acoustic model: formant positions are inversely proportional to VTL
  - Observation: formant frequencies for women are 20% higher than for men (on average)

Vocal Tract Length Normalization (VTLN)

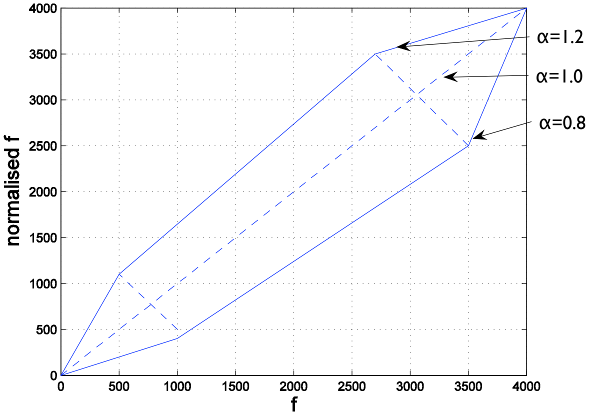
- Basic idea Normalize the acoustic data to take account of changes in vocal tract length
- Vocal tract length (VTL):
  - First larynx descent in first 2-3 years of life
  - VTL grows according to body size, and is sex-dependent
  - Puberty: second larynx descent for males
- VTL has large effect on the spectrum
  - Tube acoustic model: formant positions are inversely proportional to VTL
  - Observation: formant frequencies for women are 20% higher than for men (on average)
- VTLN: compensate for differences between speakers via a warping of the frequency axis

Approaches to VTLN

$$f \to \hat{f} = g_\alpha(f)$$

- Classify by frequency warping function
  - Piecewise linear
  - Power function
  - Bilinear transform
- Classify by estimation of warping factor α
  - Signal-based: estimated directly from the acoustic signal, through explicit estimation of formant positions
  - Model-based: maximize the likelihood of the observed data given acoustic models and a transcription. α is another parameter set so as to maximize the likelihood

Warping functions: Piecewise linear



$$\hat{f} = \alpha f$$

Model-based VTLN

- Basic idea Warp the acoustic features (for a speaker) to better fit the models — rather than warping the models to fit the features!

Model-based VTLN

- Basic idea Warp the acoustic features (for a speaker) to better fit the models — rather than warping the models to fit the features!
- Estimate the warping factor α so as to maximise the likelihood of the acoustic models

Model-based VTLN

- Basic idea Warp the acoustic features (for a speaker) to better fit the models — rather than warping the models to fit the features!
- Estimate the warping factor α so as to maximise the likelihood of the acoustic models
- After estimating the warp factors, normalize the acoustic data and re-estimate the models
- The process may be iterated

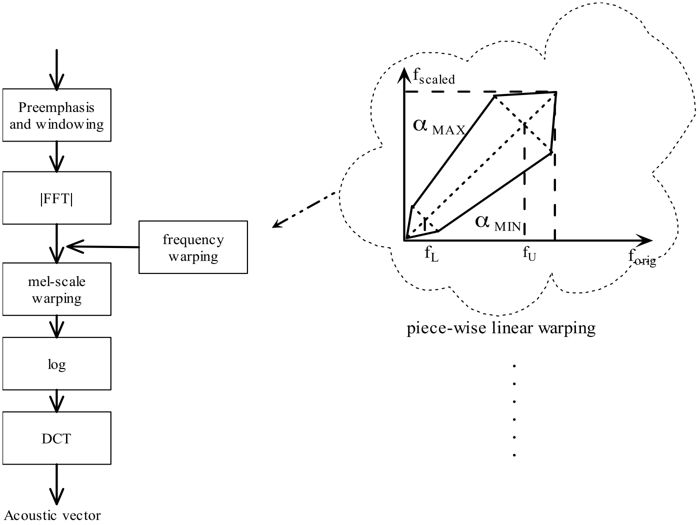
Model-based VTLN

- Basic idea Warp the acoustic features (for a speaker) to better fit the models — rather than warping the models to fit the features!
- Estimate the warping factor α so as to maximise the likelihood of the acoustic models
- After estimating the warp factors, normalize the acoustic data and re-estimate the models
- The process may be iterated
- Model-based VTLN does not directly estimate vocal tract size, rather it estimates an optimal frequency warping, which may be affected by other factors (eg $F0$)

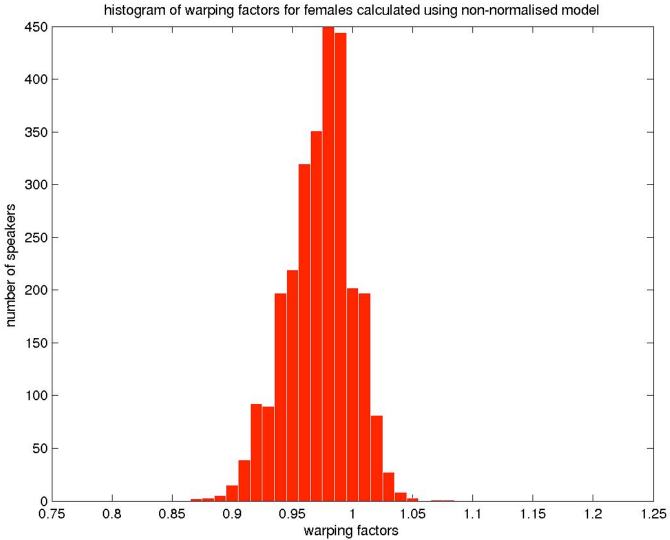
Model-based VTLN

- Basic idea Warp the acoustic features (for a speaker) to better fit the models — rather than warping the models to fit the features!
- Estimate the warping factor α so as to maximise the likelihood of the acoustic models
- After estimating the warp factors, normalize the acoustic data and re-estimate the models
- The process may be iterated
- Model-based VTLN does not directly estimate vocal tract size, rather it estimates an optimal frequency warping, which may be affected by other factors (eg $F0$)
- Exhaustive search for the optimal warping factor would be expensive
  - Approximate the log likelihood wrt α as a quadratic, and find the maximum using a line search (Brent's method)

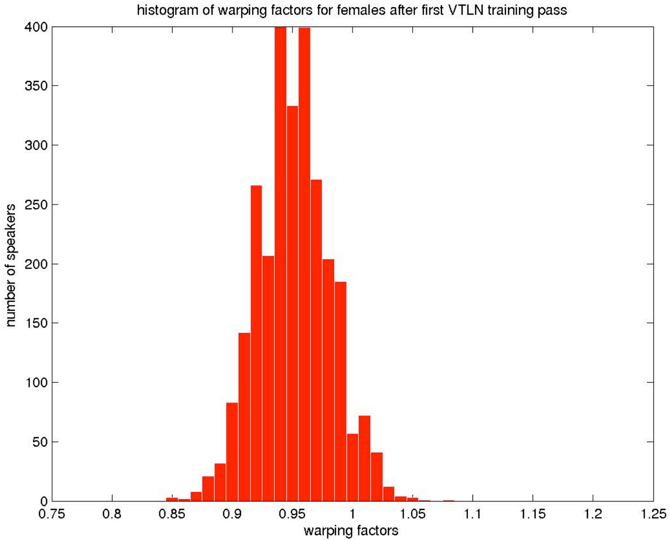
Model-based VTLN



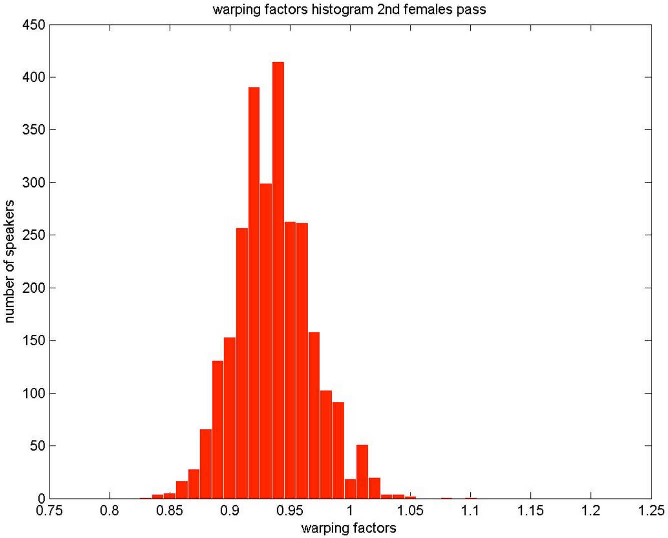
VTLN: Warp factor estimation, females, non-normalized



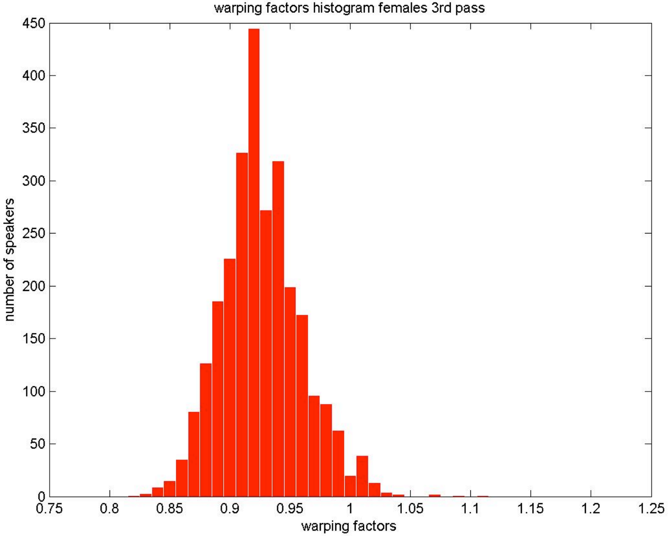
VTLN: Warp factor estimation, females, pass 1



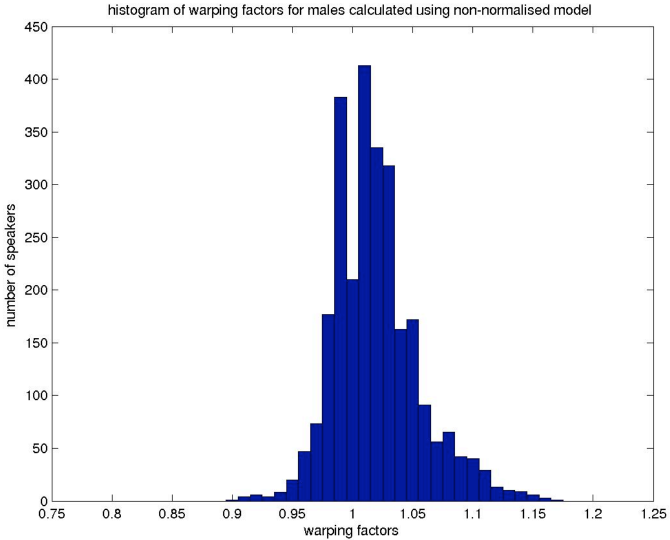
VTLN: Warp factor estimation, females, pass 2



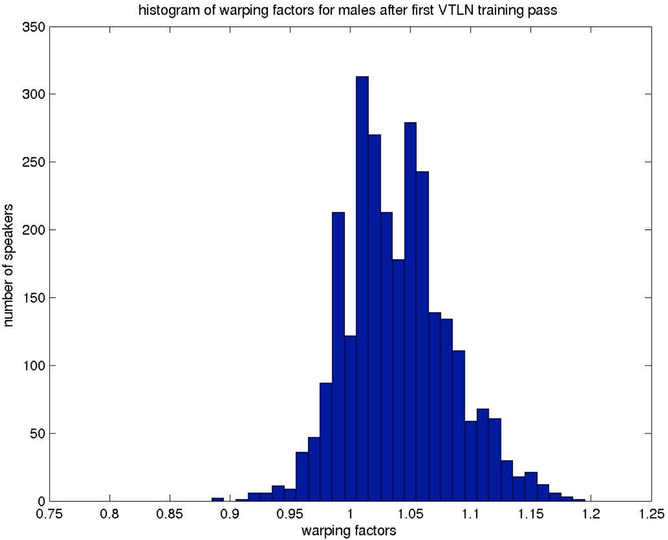
VTLN: Warp factor estimation, females, pass 3



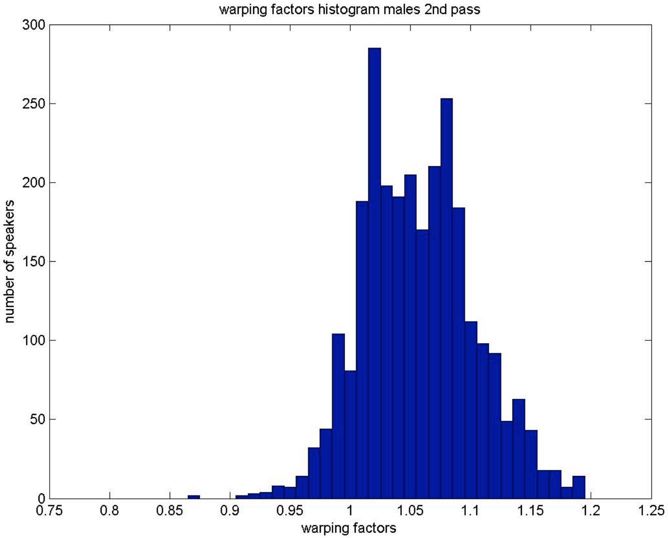
VTLN: Warp factor estimation, males, non-normalized



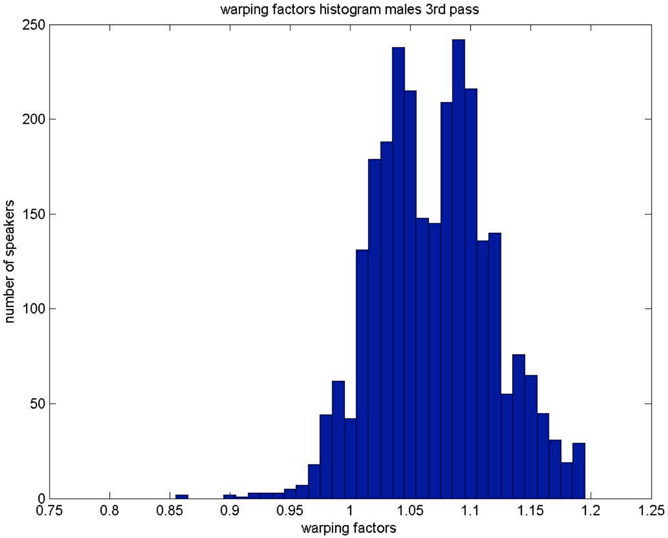
VTLN: Warp factor estimation, males, pass 1



VTLN: Warp factor estimation, males, pass 2



VTLN: Warp factor estimation, males, pass 3



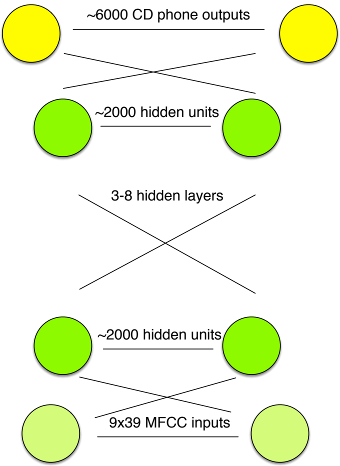
Speaker adaptation in hybrid HMM/NN systems: CMLLR feature transformation

- Basic idea: If HMM/GMM system is used to estimate a single constrained MLLR adaptation transform, this can be viewed as a feature space transform
- Use the HMM/GMM system with the same tied state space to estimate a single CMLLR transform for a given speaker, and use this to transform the input speech to the DNN for the target speaker
- Can operate unsupervised (since the GMM system estimates the transform)
- Limited to a single transform (regression class)

Speaker adaptation in hybrid HMM/NN systems: LIN – Liniear Input Network

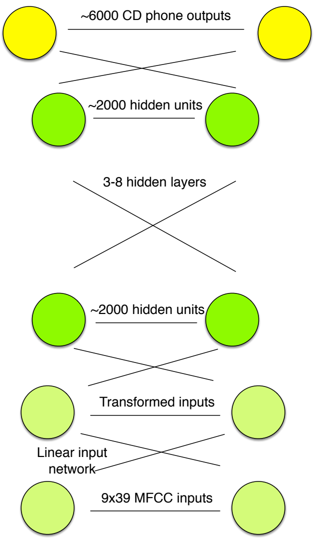
- Basic idea: single linear input layer trained to map input speaker-dependent speech to speaker-independent network
- Training: linear input network (LIN) can either be fixed as the identity or (adaptive training) be trained along with the other parameters
- Testing: freeze the main (speaker-independent) network and propagate gradients for speech from the target speaker to the LIN, which is updated — linear transform learned for each speaker
- Requires supervised training data

LIN

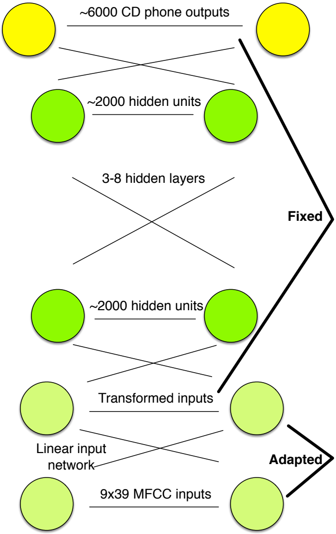


~6000 CD phone outputs
~2000 hidden units
3-8 hidden layers
~2000 hidden units
9x39 MFCC inputs

LIN


~6000 CD phone outputs
~2000 hidden units
3-8 hidden layers
~2000 hidden units
Transformed inputs
Linear input
network
9x39 MFCC inputs

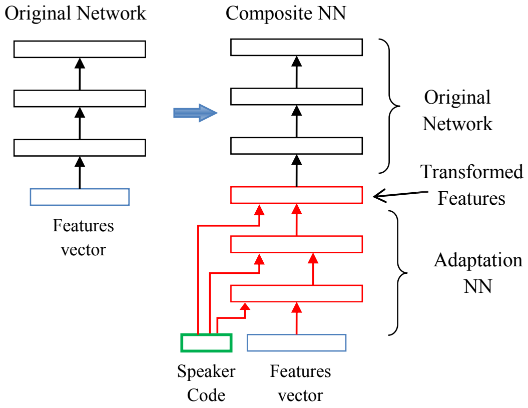
LIN



~6000 CD phone outputs
~2000 hidden units
3-8 hidden layers
Fixed
~2000 hidden units
Transformed inputs
Linear input network
Adapted
9x39 MFCC inputs

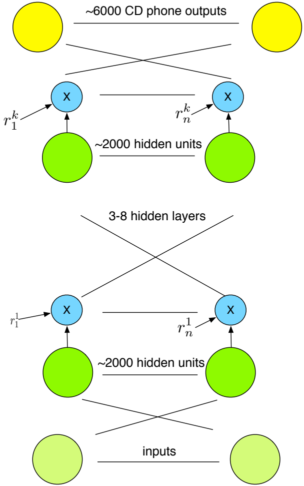
Speaker adaptation in hybrid HMM/NN systems: Speaker codes

- Basic idea: Learn a short speaker code vector for each talker

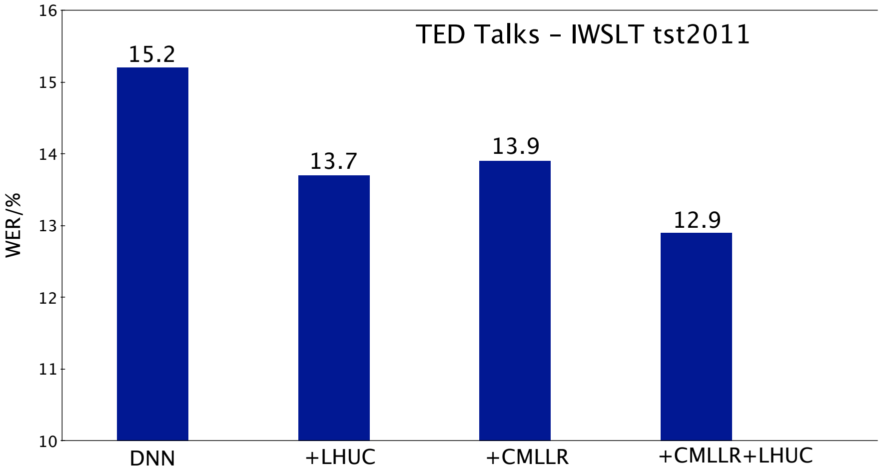


Speaker adaptation in hybrid HMM/NN systems: LHUC – Learning Hidden Unit Contributions

- Basic idea: Add a learnable speaker dependent ampolitude to each hidden unit
- Speaker independent: amplituides set to 1
- Speaker dependent: learn amplitudes from data, per speaker



Speaker adaptation in hybrid HMM/NN systems: Experimental Results on TED



Summary

Speaker Adaptation

- One of the most intensive areas of speech recognition research since the early 1990s
- HMM/GMM
  - Substantial progress, resulting in significant, additive, consistent reductions in word error rate
  - Close mathematical links between different approaches
  - Linear transforms at the heart of many approaches
- HMM/NN
  - Open research topic
  - GMM-based feature space transforms somewhat effective
  - Direct weight adaptation less effective

Reading

- HMM/GMM
  - Gales and Young (2007), The Application of Hidden Markov Models in Speech Recognition, *Foundations and Trends in Signal Processing*, **1** (3), 195–304: section 5.
  - Woodland (2001), Speaker adaptation for continuous density HMMs: A review, ISCA ITRW on Adaptation Methods for Speech Recognition.
  - Gales (1998), Maximum likelihood linear transformations for HMM-based speech recognition, *Computer Speech and Language*, **12**:75–98.
- HMM/DNN
  - Liao (2013), Speaker adaptation of context dependent deep neural networks, Proc IEEE ICASSP
  - Abdel-Hamid and Jiang (2013), Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code, Proc IEEE ICASSP
  - Swietojanski and Renals (2014), Learning Hidden Unit Contributions for Unsupervised Speaker Adaptation of Neural Network Acoustic Models, Proc IEEE SLT