Progress in Adaptation of Deep Neural Network Acoustic Models

Penny Karanasou[1], Yulan Liu[2],
Pawel Swietojanski[3], Chunyang Wu[1], **Zhinzheng Wu**[3]

1:University of Cambridge, 2:The University of Sheffield,
3:The University of Edinburgh

The University Of Sheffield.

THE UNIVERSITY *of* EDINBURGH

Natural Speech Technology

NST annual meeting - 28 May 2015

Overview

- Challenges of adaptation of DNN acoustic models
- Classification and comparison of adaptation techniques
- NST work for ASR and TTS
  - Informative priors for i-vectors estimation (P. Karanasou)
  - Speaker informed training (Y. Liu)
  - Adapting Hidden Units of DNNs (P. Swietojanski)
  - Multi-basis Adaptive DNNs (C. Wu)
  - Adaptation for TTS (Z. Wu)

Adaptation of DNN acoustic models

- Increasing interest in adaptation of DNN acoustic models
- Cases to handle: multiple speakers, different environments, channel variation...
- Challenges to address
  - Large number of parameters requires large amount of adaptation data (overfitting problem) [Liao, ICASSP2013]
  - Difficult to find structure in weights of DNN and apply transforms as in GMM-HMMs [Liao, ICASSP2013], [Seide, ASRU2011]
  - Need for small size of transforms: quick adaptation and small storage requirements
  - Joint optimisation of DNN and adaptation parameters

A classification of adaptation techniques of DNN acoustic models

- Transformation of the acoustic features
- Speaker information added as auxiliary input features
- Model-based adaptation
- Modification of the DNN structure

Comparison of adaptation methods

Transformation of acoustic features

- Speaker transformation at feature level, such as CMLLR [Gales, 1998]
- Transformation learned independently of DNN - no need for back-propagation
- Need an HMM-GMM system to generate the SAT features
- Supervised adaptation: Need to generate transcriptions if they are not available

Comparison of adaptation methods

Transformation of acoustic features

- Speaker transformation at feature level, such as CMLLR [Gales, 1998]
- Transformation learned independently of DNN - no need for back-propagation
- Need an HMM-GMM system to generate the SAT features
- Supervised adaptation: Need to generate transcriptions if they are not available

Speaker auxiliary input features

- Add speaker information to DNN input
- Small number of parameters to adapt
- Can be applied to short segments for low-latency adaptation
- Independent of DNN structure
- Not jointly estimated with DNN parameters

Comparison of adaptation methods

Model-based adaptation

- "Tune" the DNN parameters to particular speakers
- Optimize all DNN parameters jointly and discriminatively
- Large number of parameters to estimate with back-propagation

Comparison of adaptation methods

Model-based adaptation

- "Tune" the DNN parameters to particular speakers
- Optimize all DNN parameters jointly and discriminatively
- Large number of parameters to estimate with back-propagation

Modification of DNN structure

- Introduce meaningful structures in DNN for adaptation
- Small number of parameters to adapt
- Two passes of DNN training needed for adaptation (first SI, then adapted DNN)
- Usually expands the DNN structure and introduces more parameters to DNN training

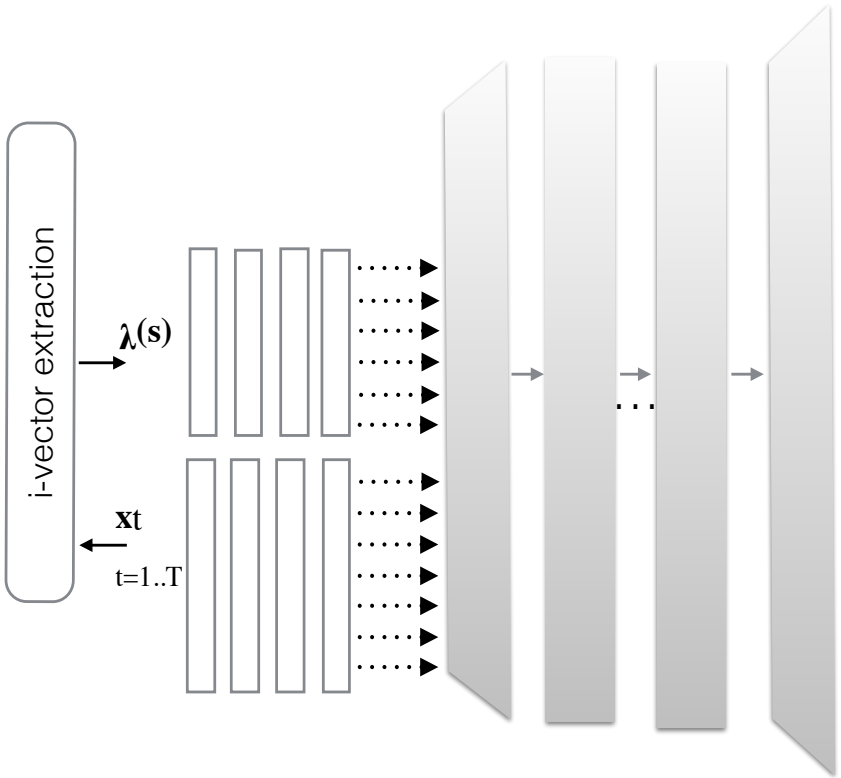
Speaker auxiliary input features

- Append speaker informed features to the input of the network
- Examples: i-vectors [Saon, ASRU2013], speaker codes [Bridle, NIPS1990] [Abdel-Hamid, ICASSP2013] , speaker separation bottleneck features (SSBN) [Liu, ICASSP2014]
- Speaker and factorised (speaker/environment) i-vectors added to the input [Karanasou, IS2014]
- Informative priors introduced to estimation of speaker i-vectors [Karanasou, submitted to IS2015]
- Establish a common theoretical framework for speaker informed DNN training methods; investigate its relationship to DNN parameters as well as input features [Liu, ICASSP15],[Liu, submitted to IS15]

I-Vector Estimation Using Informative Priors for Adaptation of DNNs

Penny Karanasou, Mark Gales, Phil Woodland

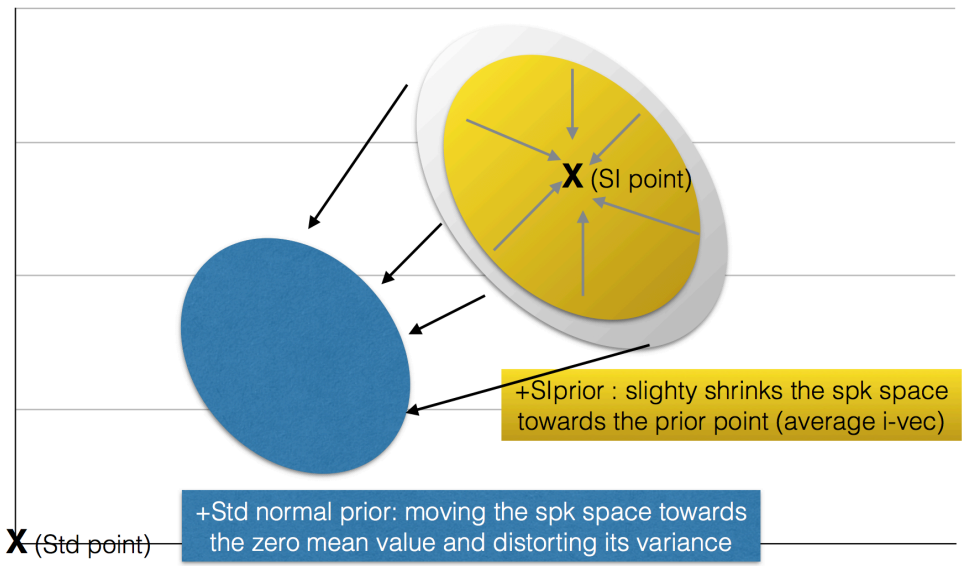
Adaptation of a hybrid DNN-HMM system with speaker i-vectors



- i-vectors: low-dimensional representation of speaker space
- A small number of parameters to estimate => can be used with very little data
- Prior needed to improve robustness of i-vector estimates with limited data

Prior-enhanced i-vectors

- Count-smoothing prior: Interpolate basic accumulates with priors estimated on the training data
- ID prior : i-vector $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- SI prior : $\boldsymbol{\lambda}_{SI}^{(s)} = \mathbf{G}_{\lambda(\mathtt{SI})}^{-1} \mathbf{k}_{\lambda(\mathtt{SI})}$



ASR performance on US BN-E corpus

Table : Hybrid decoding results for DNNs with SI input features (WER %)

System	dev03-manual
Baseline	12.7
+iv-utter	**11.5**
+iv-utter-Stdprior	14.2
+iv-utter-SIprior	11.5
+iv-utter-Stdprior-retrain	11.6
+iv-utter-SIprior-trn-retrain	**11.1**

- "+iv-utter": append utterance-level test i-vectors to DNN input
- Compare "+iv-utter-Stdprior", "+iv-utter-Stdprior-retrain": Std normal prior sensitive to mismatch of trn/test i-vector spaces
- Best performance with utter-level test i-vectors with informative prior ("+iv-utter-SIprior-retrain")

Speaker Informed DNN Training

Yulan Liu, Thomas Hain

Speaker Informed DNN & Bias Adaptation

- Speaker informed DNN training can be equivalent to bias adaptation at the input layer
  - Particularly, using speaker based auxiliary codes is equivalent to using speaker dependent biases.
- The design of auxiliary codes influences the adaptation performance
  - Particularly the dimension, discriminability and stability of the auxiliary codes;
  - With a proper design, hand-crafted codes can achieve equivalent performance with i-vectors.

Speaker Informed DNN & Input Normalisation

- Speaker informed DNN training can be equivalent to speaker based additive DNN input normalisation
  - Factorise speaker dependent biases linearly but in different structures;
  - Performance of two methods can be equivalent, while combining them does not improve further;
  - Additive input normalisation over log filter bank features enables an interpretation of speaker dependent scaling over spectrum.
- Other normalisation methods
  - Speaker based multiplicative input normalisation is also effective, however additive normalisation wins out;
  - Combining additive and multiplicative normalisations brings marginal further improvement;
  - Joint optimisation with DNN parameters is crucial in input normalisation.

Model-based adaptation

- Find a way to "tune" the DNN parameters to particular speakers
- Factorise hidden layer(s) and update smaller matrices only [Xue, IS2014] or update output layer only [Yao, SLT2012]
- Reguralise any type of model with KL-like criterion so the adapted model does not diverge too much from its unadapted version [Yu, ICASSP2013]
- Scale hidden units using speaker-dependent data (Learning Hidden Units Contributions -LHUC) [Swietojanski, SLT2014]
- Hidden units interpolation within pooling regions [Swietojanski, ICASSP 2015]

Adapting Hidden Units of Neural Networks for Acoustic Modelling

Pawel Swietojanski, Steve Renals

Learning Hidden Unit Contributions (LHUC)

- Each hidden unit states some hypothesis $\mathcal{H}_i$ (defined by its parametrisation θ_i) about its inputs (data) $\mathbf{x}$, i.e. for the ith hidden unit and sigmoid activation (ϕ) one can write:

$$h_i^{l+1} = \phi(\mathbf{x}\mathbf{W} + \mathbf{b}) = P(\mathcal{H}_i|\mathbf{x}; \theta_i)$$

- The set of hypotheses in the model is structured (into layers) and jointly optimised during training (but do their relative importance remains optimal for unseen data?)
- LHUC re-weights the contributions of particular hidden units using adaptation data for the mth speaker, as follows:

$$h_i^{l+1} = a(r_i^m) \circ \phi(\mathbf{x}\mathbf{W} + \mathbf{b})$$

Differentiable pooling (DiffP)

- Like LHUC, but performs hidden units interpolation within pooling regions (instead of scaling)

$$h_k^{l+1} = \sum_{i \in G_k} u_i^{m,k} h_i^l$$

- where $u^{m,k}$ is some non-linear function of h_i^l and its parametrisation depends on speaker m and kth pooling unit,
- At test time, one refines only pooling parameters of $u^{m,k}$ in per-speaker manner
- Experimented with two forms of pooling, linear weighting with pooling weights defined by Gaussian kernels [Swietojanski, ICASSP2015] and L_p-norm (with learnable order p) [Under preparation].

Results - Word Error Rates (%)

Observed around 5-20% relative improvements across various corpora. Methods were found to be complementary to each other as well as to CMLLR.

Example numbers:

- LHUC
  - TED (tst2010): 14.9 $\rightarrow$ 12.9
  - Switchboard (eval2000): 22.1 $\rightarrow$ 21.2
  - Aurora4 (multi-condition): 11.8 $\rightarrow$ 9.5 (or 10.8 $\rightarrow$ 8.6 with dropout)
- DiffP
  - TED (tst2010): 14.9 $\rightarrow$ 12.9
  - Switchboard (eval2000): 21.3 $\rightarrow$ 20.3
- LHUC + DiffP
  - TED (tst2010): 14.9 $\rightarrow$ 12.5

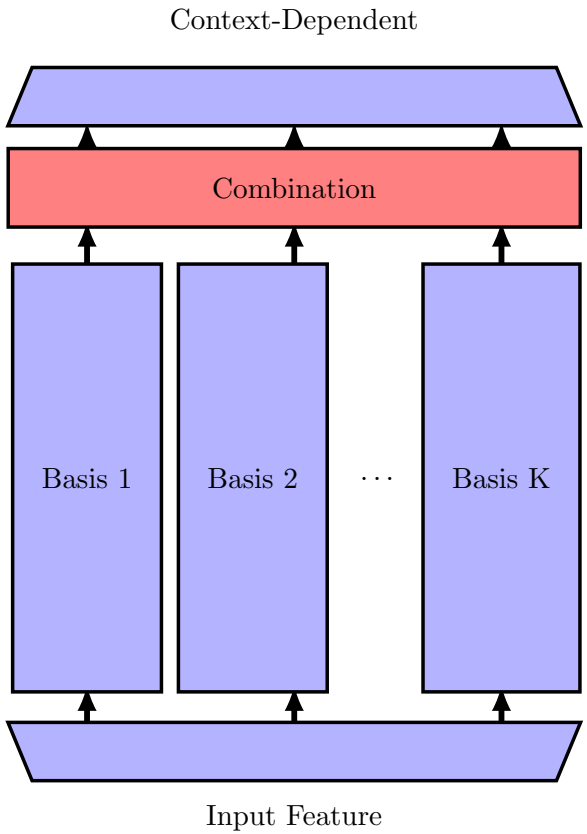
Modification of the DNN structure

- Introduce meaningful structures in DNN for adaptation, which are not explicit and hard to figure out in traditional NN parameters
- Additional linear layers as speaker-dependent transforms prior to the input layer [Neto, IS1995], to a hidden layer [Gemello, SpeechComm2007] or to the output layer [Li, IS15]
- Set of sub-networks structure (called bases) inspired by CAT [Wu, ICASSP2015] [Tan, ICASSP2015]. Adapt the DNN by learning the interpolation weights of the bases for each speaker
- Convolutional layers with frequency pooling [Abdel-Hamid, ICASSP2012]

Multi-basis Adaptive Neural Network for Rapid Adaptation in Speech Recognition

Chunyang Wu, Mark Gales

Multi-basis Adaptive Neural Network



- Introduce multiple bases
- Shared common input and output layers (Optionally common hidden layers)
- Bases are combined via interpolation

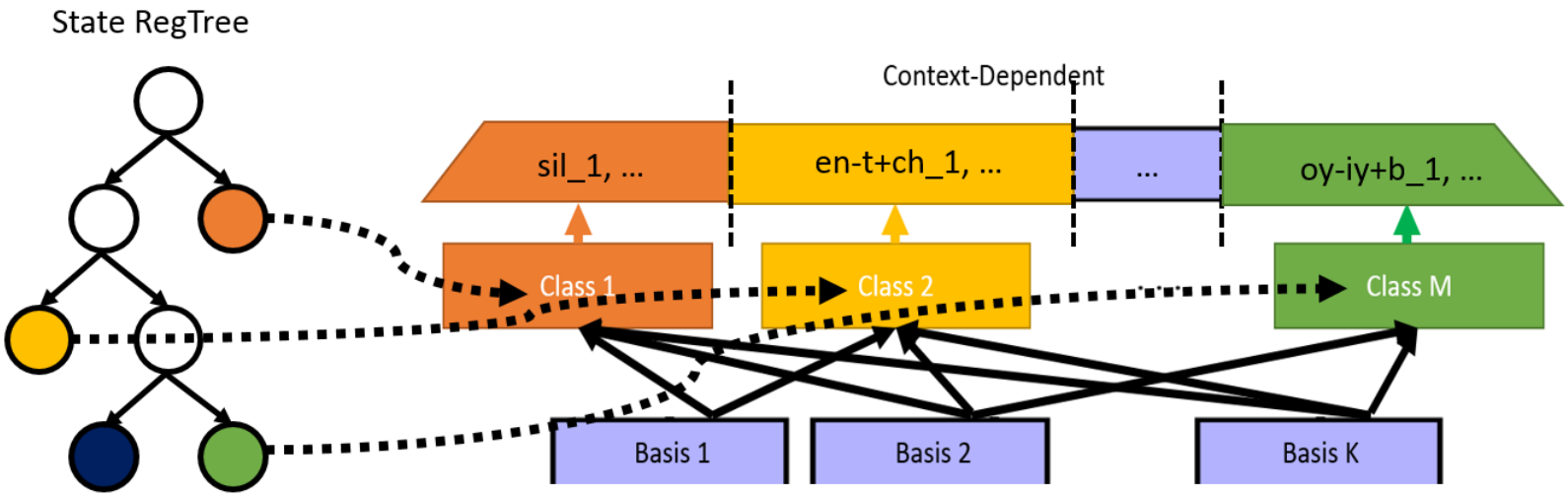
$$p(y = i|x) = \frac{\exp\left(\sum_k \lambda_k z_i(x)\right)}{\sum_j \exp\left(\sum_k \lambda_k z_j(x)\right)}$$

- $\boldsymbol{\lambda}$ **convex** optimization on the cross-entropy criterion

Extension with Multi-Interpolation Classes

- Introduce phonetic knowledge while adapting the acoustic space
- Each output (i-th CD-state) is given via interpolation on its corresponding class $c(i)$ interpolation weights

$$p(y = i|x) = \frac{\exp\left(\sum_k \lambda_k^{c(i)} z_i(x)\right)}{\sum_j \exp\left(\sum_k \lambda_k^{c(j)} z_j(x)\right)}$$



Speaker adaptation of DNN acoustic models for TTS

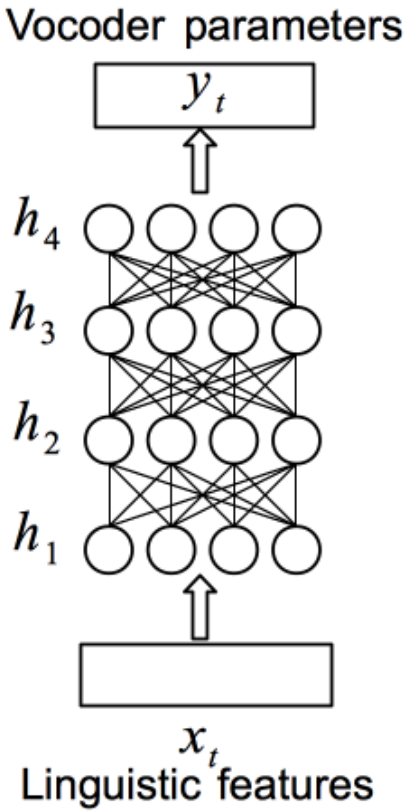
Zhizheng Wu, Simon King

Speaker adaptation for speech synthesis

- Create a new voice using a small amount of target speech and average voice model
- Adaptability is one of the major advantages of statistical parametric speech synthesis over unit selection
- Significant amount of work has been done in HMM-based speech synthesis
- Will DNN models achieve better adaptation performance than HMM?

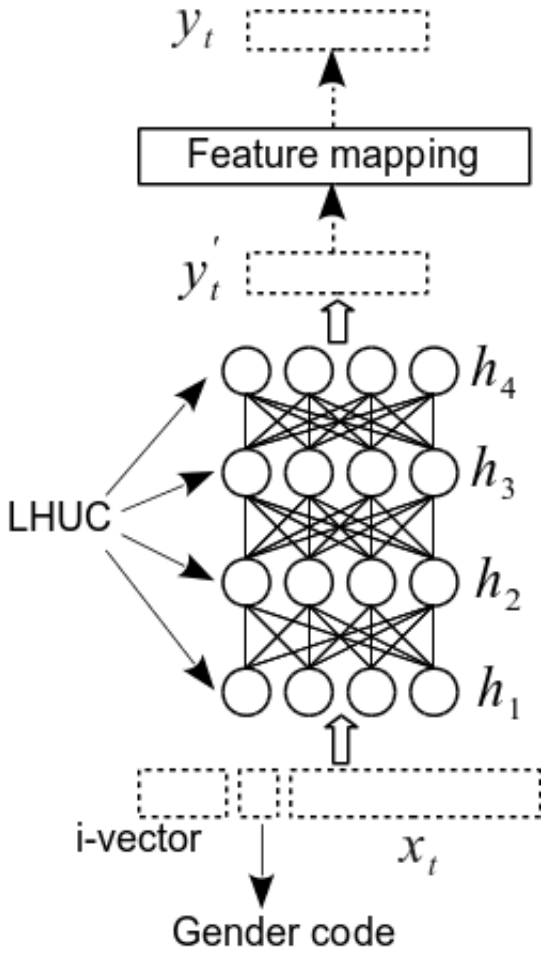
DNN-based speech synthesis

- Map linguistic features to vocoder parameters



Speaker adaptation for DNN synthesis

- Speaker adaptation can be done at three levels

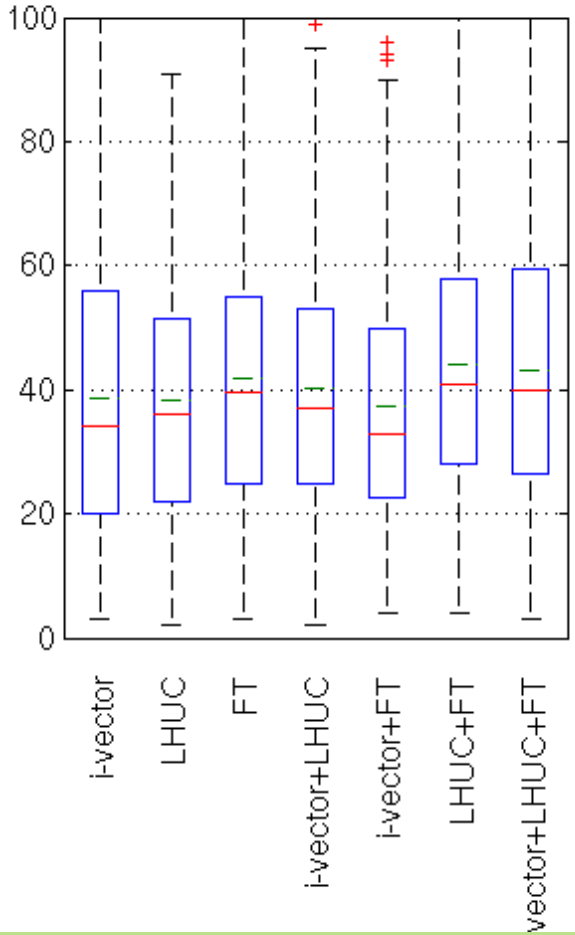


Speaker adaptation for DNN synthesis

- Input level: i-vector
- Model level: LHUC
- Output level: Feature transformation

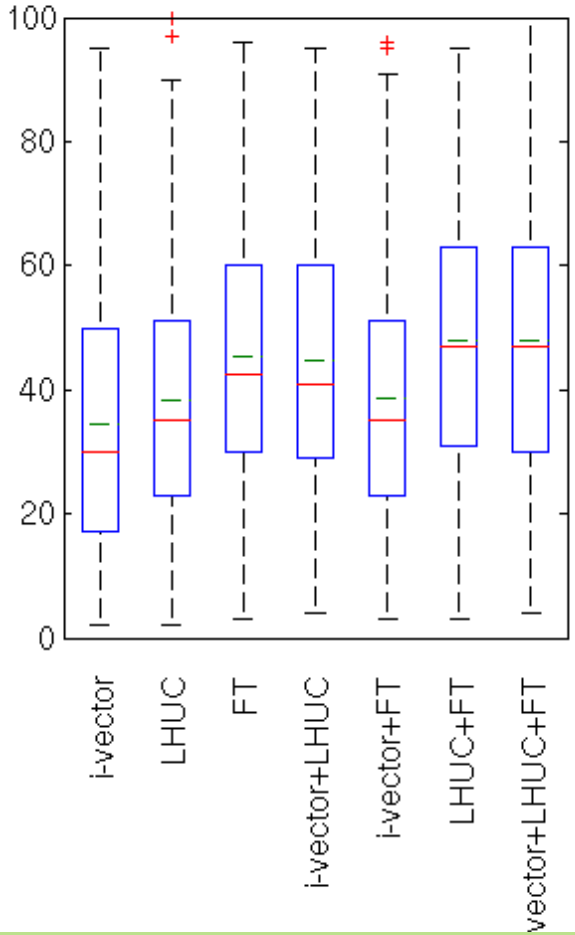
Speaker adaptation for DNN-based speech synthesis

- Naturalness: 10 utterances adaptation



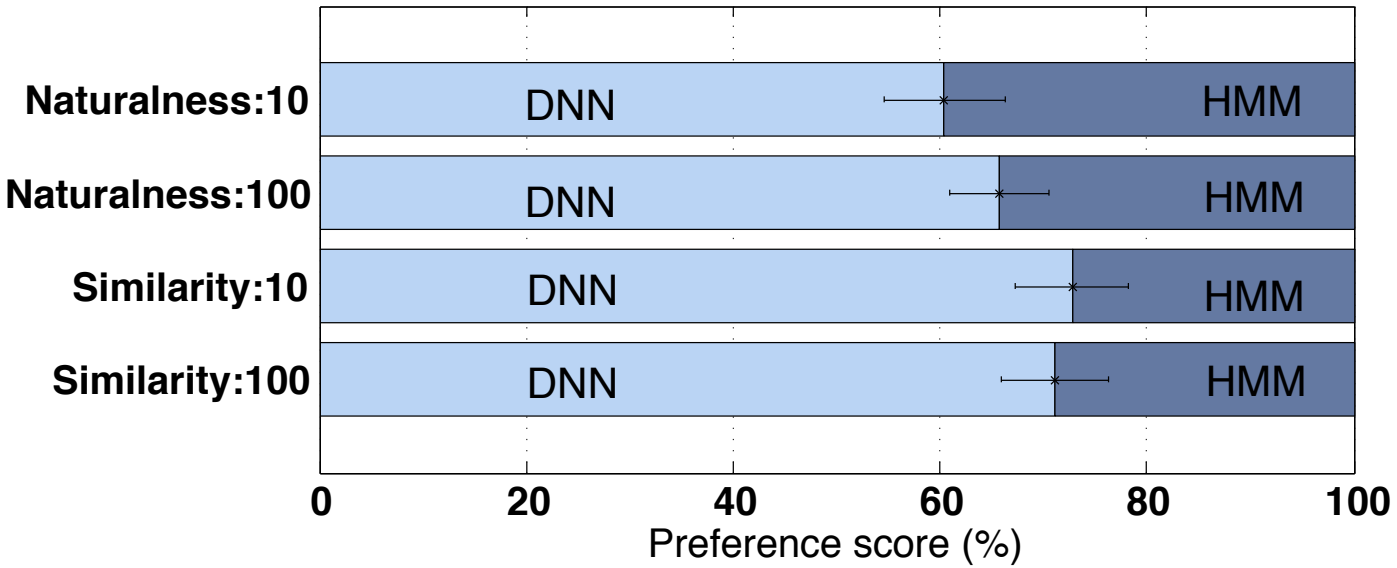
Speaker adaptation for DNN-based speech synthesis

- Similarity: 10 utterances adaptation



Speaker adaptation: DNN vs HMM

- Preference test



Thank you!