Using eigenvoices and nearest-neighbours in HMM-based cross-lingual speaker adaptation with limited data

Using Eigenvoices and Nearest-Neighbours in HMM-Based Cross-Lingual Speaker Adaptation with Limited Data

Seyyed Saeed Sarfjoo*, *Student Member, IEEE,* Cenk Demiroglu†, *Member, IEEE,* and Simon King‡, *Fellow, IEEE*
*†Electrical and Computer Engineering Department, Ozyegin University, Istanbul, Turkey
‡The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
*saeed.sarfjoo@ozu.edu.tr, †cenk.demiroglu@ozyegin.edu.tr, ‡simon.king@ed.ac.uk

***Abstract*—Cross-lingual speaker adaptation for speech synthesis has many applications, such as use in speech-to-speech translation systems. Here, we focus on cross-lingual adaptation for statistical speech synthesis systems using limited adaptation data. To that end, we propose two eigenvoice adaptation approaches exploiting a bilingual Turkish-English speech database that we collected. In one approach, eigenvoice weights extracted using Turkish adaptation data and Turkish voice models are transformed into the eigenvoice weights for the English voice models using linear regression. Weighting the samples depending on the distance of reference speakers to target speakers during linear regression was found to improve the performance. Moreover, importance weighting the elements of the eigenvectors during regression further improved the performance. The second approach proposed here is speaker-specific state-mapping which performed significantly better than the baseline state-mapping algorithm both in objective and subjective tests. Performance of the proposed state mapping algorithm was further improved when it was used with the intra-lingual eigenvoice approach instead of the linear-regression based algorithms used in the baseline system.**

Index Terms: statistical speech synthesis, speaker adaptation, nearest neighbour, cross lingual speaker adaptation

I. Introduction

Cross-lingual speaker adaptation (CLSA) for statistical speech synthesis is a method for adapting a text-to-speech (TTS) system for a desired *output* language, given adaptation data (i.e., speech) from the target speaker in a different *input* language. Applications include speech-to-speech translation [1], [2].

In a commonly used approach [3]–[5], a speaker-independent acoustic model (an "Average Voice Model" or AVM) for each of the two languages is required. A mapping between pairs of corresponding states in the two models is constructed, on the basis of the states' acoustic similarity. Then, either the adaptation data itself, or speaker transformation functions, can be mapped from the input language acoustic model to the output language acoustic model.

Mismatch between the two AVMs degrades the quality when mapping transforms [6], [7] since the speaker-specific transformations for states in the input language acoustic model may not actually suit the corresponding state in the output language acoustic model. To alleviate that problem, a transform mapping using shared decision tree context clustering is proposed in [8] where not only acoustic-similarity but also contextual similarity of states is taken into account during mapping.

The AVM can also be trained using data from multiple languages and adapted to a target speaker that speaks one of those languages [9]. However, the adaptation of such a model may hampered by the fact that some leaf nodes of the decision tree might be trained with data from only one language. A speaker and language factorization technique to alleviate this problem is proposed in [10] where Cluster Adaptive Training (CAT) is used to build an AVM using data from different languages. For a given target language, cluster weights are estimated for building a language-dependent model, before adapting it to a speaker of that language.

For decreasing language dependency and also adapting prosodic information in CLSA, the mapping between languages can be provided by a language-independent space of perceptual characteristics (PC) [11]. This technique relies on two language spaces of speakers' voices in the input and output languages. Each speaker is represented by a mean super-vector. When a new target speaker enters the input language speaker space, it is first projected to the intermediary PC space and, once an appropriate representation for this speaker is found in that space, it is projected to the output language. Finally, speaker interpolation is performed in the output language to reconstruct the super-vector of the target speaker. The perceptual space is constructed using listening tests.

Factor analysis-based CLSA using bilingual speech data is proposed in [12]. In this method, model parameters representing language-dependent acoustic features and factors representing speaker characteristics are simultaneously optimized using a maximum likelihood approach and a single statistical model trained using bilingual speech data. Assuming that the speaker characteristics factors are the same in both languages, performance is expected to improve compared to training each eigenvoice space independently.

A voice conversion algorithm is proposed in [13] for rapid cross-lingual adaptation. An eigenvoice-based conversion model is learned using parallel data between a source speaker and a pool of speakers speaking the same language

This work is supported by the EU Marie Curie programme under grant number 268409 and the TUBITAK 3501 programme under project number 109E281.

as the source speaker. Then, that model is adapted to a target speaker that speaks a foreign language using a small amount of data.

Deep neural networks methods have also been used for training multilingual acoustic models [14]–[16]. However, such models need a significant amount of data for training and adaptation whereas the focus here is adaptation with limited data.

In this paper, we focus on cross-lingual adaptation when only a few utterances are available from a target speaker. In our recent paper [17], to achieve better speaker similarity than existing state-mapping based algorithms under limited data conditions, we proposed two methods. In the first method, eigenvoices were used for rapid adaptation. Eigenvoice weights computed for the input language are linearly transformed into output language weights. The transformation matrix is learned using a bilingual training database which contains English and Turkish speech data from the same speakers.

In the second method, we proposed speaker-specific state-mapping, for which a bilingual database was used. After generating speaker-adapted models for both input and output languages, a speaker-specific state-map is constructed for each speaker in the pool of bilingual speakers. Then, for a previously-unseen target speaker, a nearest-neighbour is found in the pool and the state map of that nearest-neighbour is used for adaptation. Performance for the excitation parameters was found to be significantly better with the proposed method than the baseline target-speaker-independent state-mapping algorithm, in objective and subjective tests.

The novelty of this paper is as follows. First, we give a more detailed description of our previous work [17] with additional experimental results, such as quality tests, with more native listeners and more discussion of results. The second novelty is that during eigenvector transformation, to avoid overfitting and exploit correlations within eigenvector elements, a partial least squares (PLS) approach is used. To further boost the performance, elements of eigenvectors are also weighted using recursive PLS (rPLS). Moreover, in addition to weighting the eigenvectors in a least-squares linear regresssion approach, as done in [17], eigenvectors are weighted in the proposed PLS and rPLS frameworks leading to weighted-PLS and weighted-rPLS algorithms. As the last novelty, the proposed state-mapping algorithm is used for mapping the data in the input language to models in the output language and performing cross-lingual eigenvoice adaptation which enabled significant improvement in the spectral envelope features.

This paper is organized as follows. The baseline cross-lingual speaker adaptation method is described in Section II, and the eigenvoice approach to statistical speech synthesis (SSS) in Section III. The proposed algorithms are described in Section IV with experimental results in Section V. Finally, a conclusion is given in Section VI.

II. Baseline state mapping algorithm

State-mapping is one of the most successful methods for cross-lingual speaker adaptation [5]. In this approach, average voice models (AVMs) in the input and output languages are trained and then a mapping between pairs of states in the two models is formed, typically by finding pairs of states with the smallest Kullback-Leibler divergence (KLD) [18].

Each adaptation data vector in the input language is associated with a state in the AVM of that language using forced alignment of AVM states with the data vectors. The data can then be mapped to the corresponding state in the output language AVM using the mapping between the input and output language AVM states. Once the adaptation data vectors are mapped to states in the output language AVM, they can be used to adapt the output language AVM parameters using any intralingual adaptation method such as constrained maximum likelihood linear regression (CMLLR) [19], [20], constrained structural maximum a posteriori linear regression (CSMAPLR) [21] or vocal tract length normalization (VTLN) [22].

Alternatively, the adaptation transforms can be learned with respect to the input language AVM and then used to transform the parameters of the corresponding states in the output language AVM. Whilst the data mapping approach achieves better speaker similarity, the transform mapping approach achieves better speech quality [5]. Because our focus is on improving speaker similarity, we employ the data mapping approach in the baseline system.

III. Eigenvoice adaptation

With very limited adaptation data, an eigenvoice approach can be used [23], [24]. Given a set of R eigenvectors $\boldsymbol{e}_r \in \mathbb{R}^n$, which are called eigenvoices in this context, the mean supervector for speaker s is $\boldsymbol{\mu}^{(s)} = [\boldsymbol{\mu}_1^{(s)^T}\ \boldsymbol{\mu}_2^{(s)^T}\ ...\ \boldsymbol{\mu}_{N_{st}}^{(s)^T}]^T$ where N_{st} is the total number of states in the acoustic model, and $\boldsymbol{\mu}_c^{(s)}$ is the mean vector of the c^{th} state; $\boldsymbol{\mu}^{(s)}$ can be modeled as:

$$\boldsymbol{\mu}^{(s)} = \boldsymbol{\mu}_{\text{SI}} + \boldsymbol{E}\boldsymbol{w}_s + \boldsymbol{\epsilon}_s \tag{1}$$

where $\boldsymbol{\mu}_{\text{SI}}$ is the mean supervector of the AVM (i.e., a speaker-independent model), $\boldsymbol{E} = [\boldsymbol{e}_1\ \boldsymbol{e}_2\ ...\ \boldsymbol{e}_R]$ is a matrix of eigenvectors spanning the space of speakers in the AVM, $\boldsymbol{w}_s$ is the weight vector for speaker s, and $\boldsymbol{\epsilon}_s$ is the approximation error.

To perform cross-lingual speaker adaptation, we use Principal Component Analysis (PCA) to estimate $\boldsymbol{E}_{in}$ and $\boldsymbol{E}_{out}$ for the input and output language AVMs respectively. A maximum-likelihood approach is then used for estimating $\boldsymbol{w}_s$ as follows. Given some adaptation data $\boldsymbol{\chi}_a = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N_{o,s})}\}$, where $N_{o,s}$ is the total number of observations (i.e., frames) from speaker s, the likelihood function is

$$p(\boldsymbol{\chi}_a|\boldsymbol{w}_s, \boldsymbol{E}) \propto exp\Bigg(-\frac{1}{2} \sum_{c=1}^{N_{st}} \sum_{i=1}^{N_c^{(s)}} (\boldsymbol{x}_c^{'(i)} - \boldsymbol{E}_c\boldsymbol{w}_s)^T \boldsymbol{\Sigma}_c^{-1} (\boldsymbol{x}_c^{'(i)} - \boldsymbol{E}_c\boldsymbol{w}_s) \Bigg) \tag{2}$$

where $\boldsymbol{E}_c \in \mathbb{R}^{F\times R}$ is the c^{th} block of the $\boldsymbol{E}$ matrix corresponding to state c, and F is the size of the mean vectors.

$\boldsymbol{x}_c^{'(i)} = \boldsymbol{x}_c^{(i)} - \boldsymbol{\mu}_c$, $\boldsymbol{x}_c^{(i)}$ is the i^{th} observation that is aligned with state c, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the speaker-independent mean vector and covariance matrix of the Gaussian emission pdf of state c, and $N_c^{(s)}$ is the total number of observations aligned with state c for speaker s. Here, Viterbi alignment is used for likelihood estimation.

The weight vector of speaker s, $\boldsymbol{w}_s \in \mathbb{R}^{R\times 1}$, is estimated as

$$\hat{\boldsymbol{w}}_s = \boldsymbol{G}_w^{(s)^{-1}} \boldsymbol{k}_w^{(s)} \tag{3}$$

where

$$\boldsymbol{G}_w^{(s)} = \sum_{c=1}^{N_{st}} N_c^{(s)} \boldsymbol{E}_c^T \boldsymbol{\Sigma}_c^{-1} \boldsymbol{E}_c \tag{4}$$

$$\boldsymbol{k}_w^{(s)} = \sum_{c=1}^{N_{st}} \boldsymbol{E}_c^T \boldsymbol{\Sigma}_c^{-1} \boldsymbol{S}_{x,c}^{(s)} \tag{5}$$

$$\boldsymbol{S}_{x,c}^{(s)} = \sum_{i=1}^{N_c^{(s)}} \boldsymbol{x}_c^{'(i)} \tag{6}$$

Because our focus here is on adaptation with limited data, regularization during weight estimation to avoid overfitting becomes important. Thus, as opposed to using the maximum-likelihood solution in Eq. 3, we use the regularized solution described below.

Regularization is done by imposing a zero-mean Gaussian prior, $p(\boldsymbol{w})$, on the weight vector. $\boldsymbol{w}_s$ is then estimated using a maximum a posteriori (MAP) adaptation.

In the MAP approach, the weight vector for a target speaker s is estimated with the objective function

$$\hat{\mathbf{w}}_{\text{s,map}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\boldsymbol{\chi}_a|\mathbf{w})\mathbf{p}(\mathbf{w}) \tag{7}$$

where $p(\mathbf{w})$ is the prior, set to $\mathcal{N}(0, \boldsymbol{\Sigma}_{\mathbf{w}})$ here.

Using Eq (2) to replace the likelihood term $p(\boldsymbol{\chi}_a|\mathbf{w_s})$, removing the terms that are independent of $\mathbf{w}$ from the objective function, and with some matrix manipulation, the MAP objective function becomes

$$\hat{\mathbf{w}}_{\text{s,map}} = \underset{\mathbf{w}}{\operatorname{argmax}} \exp(\mathbf{w^T E^T \Sigma^{-1} S_x} - \frac{1}{2}\mathbf{w^T E^T N \Sigma^{-1} E w}) \exp(-\frac{\mathbf{1}}{\mathbf{2}}\mathbf{w^T \Sigma_w^{-1} w}), \tag{8}$$

where the block diagonal $\boldsymbol{\Sigma}^{-1} = \mathbf{diag}(\boldsymbol{\Sigma}_1^{-1}, \boldsymbol{\Sigma}_2^{-1}, ..., \boldsymbol{\Sigma}_{\mathbf{N_{st}}}^{-1})$, $\mathbf{S_x} = [\mathbf{S_{x,1}}, \mathbf{S_{x,2}}, ..., \mathbf{S_{x,N_{st}}}]$, and $\mathbf{N} = \mathbf{diag}(\mathbf{N_1}, \mathbf{N_2}, ..., \mathbf{N_{N_{st}}})$.

The objective function can be maximized by noting that the posterior distribution $p(\mathbf{w}|\boldsymbol{\chi}_\mathbf{a})$ is a Gaussian since the Gaussian distribution is the conjugate prior of the Gaussian likelihood function with unknown mean in Eq (2). Therefore, Eq (7) can be written as

$$\hat{\mathbf{w}}_{\text{s,map}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, exp(-\frac{1}{2}(\mathbf{w}-\mu_{\mathbf{w}|\chi})^{\mathbf{T}}\mathbf{R}_{\mathbf{w}|\chi}(\mathbf{w}-\mu_{\mathbf{w}|\chi})) \tag{9}$$

where $\mathbf{R}_{\mathbf{w}|\chi}$ is the precision matrix. By completing the squares and using Eq (8),

$$\mathbf{R}_{\mathbf{w}|\chi} = (\mathbf{E^T N \Sigma^{-1} E} + \boldsymbol{\Sigma}_{\mathbf{w}}^{-\mathbf{1}}), \tag{10}$$

and

$$\mu_{\mathbf{w}|\chi} = \mathbf{R}_{\mathbf{w}|\chi}^{-\mathbf{1}} \mathbf{E^T \Sigma^{-1} S_x}. \tag{11}$$

MAP estimate of $\mathbf{w}$, $\hat{\mathbf{w}}_{\text{s,map}}$, is the mean, $\mu_{\mathbf{w}|\chi}$, of the posterior distribution. $\boldsymbol{\Sigma}_{\mathbf{w}}^{-\mathbf{1}}$ is a hyper-parameter of the prior which we set to $\alpha\mathbf{S}^{-\mathbf{1}}$ where α is a scalar (chosen empirically) and $\mathbf{S}$ is the diagonal matrix

$$\mathbf{S} = \text{diag}(\lambda_\mathbf{1}, \lambda_\mathbf{2}, ..., \lambda_\mathbf{R}) \tag{12}$$

where λ_i are the eigenvalues obtained while estimating the E matrix using PCA.

Because adaptation data is available only in the input language, the computations above perform intra-lingual adaption: that is, they result in an estimate for $\boldsymbol{w}_{s,in}$. However, the weight vector for the output language, $\boldsymbol{w}_{s,out}$, is required for cross-lingual adaptation, so that we can compute

$$\boldsymbol{\mu}_{out}^{(s)} = \boldsymbol{\mu}_{\text{SI,}out} + \boldsymbol{E}_{out}\boldsymbol{w}_{s,out}. \tag{13}$$

where $\boldsymbol{\mu}_{out}^{(s)}$ is the supervector, $\boldsymbol{E}_{out}$ is the eigenvoice matrix, and $\boldsymbol{\mu}_{SI,out}$ is the speaker-independent supervector for the output language. We have investigated both data-mapping and vector-/space-mapping techniques to estimate $\boldsymbol{w}_{s,out}$. Our proposed techniques are described below.

IV. Cross-lingual eigenvoice adaptation

A. Algorithms based on eigenvector mapping

Given $\boldsymbol{w}_{s,in}$, computed using intra-lingual adaptation, we can use linear regression to predict $\boldsymbol{w}_{s,out}$. The $\boldsymbol{w}_s$ vectors for a set of bilingual training speakers can be computed for the input and output languages using Eq (3). Then, a linear regression matrix $\boldsymbol{A}$ can be trained such that $\boldsymbol{w}_{s,out} = \boldsymbol{A}\boldsymbol{w}_{s,in} + \boldsymbol{\epsilon}$. In the simplest approach, the least-squares (LS) algorithm is used for training $\boldsymbol{A}$. Once A is trained using the training speaker pool, it can be used to transform the eigenvoice weight vector of a target speaker in input language space into a vector in output language space.

Because the relationship between the input and output vectors is not linear and the number of bilingual speakers is not large, more sophisticated regression techniques are investigated and described below.

1) Speaker-specific Regression of Eigenvoice Vectors: A linear model is chosen because nonlinear methods (e.g., neural networks) require significantly more data, and collection of large bilingual databases is expensive. However, to improve the performance of the linear model, the $\boldsymbol{A}$ matrix can be constructed in a target-specific manner. To that end, we propose a weighted linear regression approach as described below.

Given adaptation data from a target speaker, the speaker-specific $\boldsymbol{A}_{tar}$ matrix is computed using:

$$\boldsymbol{A}_{tar} = \underset{\boldsymbol{A}}{\operatorname{argmin}} \sum_{i=1}^{N_p} \boldsymbol{\epsilon}_{i,tar}^T \boldsymbol{\epsilon}_{i,tar} \tag{14}$$

where N_p is number of training speakers and

$$\boldsymbol{\epsilon}_{i,tar} = L_{tar}(i).(\boldsymbol{w}_{out}(i) - \boldsymbol{A}\boldsymbol{w}_{in}(i)) \tag{15}$$

where $L_{tar}(i)$ is the weight of the i^{th} training speaker, $\boldsymbol{w}_{out}(i)$ is its eigenvoice vector in the output language and $\boldsymbol{w}_{in}(i)$ is its eigenvoice vector in the input language.

The speaker weights $L_{tar}(i)$ are computed as follows. First, intra-lingual adaptation is done and the distance of the target speaker to each of the training speakers is found by using the Euclidean (L_2) distance between the mean supervectors. Then, these distances are compressed and normalized with

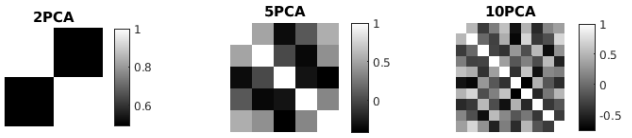
$$L_{tar}(i) = 1 - \log_2\left(\frac{d(i) - d_{min}}{d_{max} - d_{min}} + 1\right) \tag{16}$$

where $d(i)$ is the distance of the i^{th} training speaker to the target, d_{max} is the maximum and d_{min} the minimum of such distances across all training speakers.

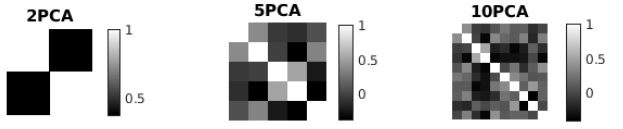
Once $L_{tar}(i)$ and A_{tar} are computed, the eigenvoice weight in the output language is estimated as $\hat{\boldsymbol{w}}_{tar,out} = \boldsymbol{A}_{tar}\boldsymbol{w}_{tar,in}$.

2) Partial Least-squares Regression: Because the number of bilingual speakers is, as already noted, not large, overfitting can occur during linear regression, especially if the eigenvoice vector dimension is large. Correlations between the elements of the eigenvoice vectors, as shown in Figure 1, can be exploited to avoid poor generalization.

When significant co-linearity exists, one way to address the overfitting problem is to use PCA and reduce the dimension of the eigenvoice vectors. However, this is not desirable in our case because the linear regression step is already preceded by a PCA step and further reduction of dimensionality would cause degradation in adaptation performance. Moreover, PCA only minimizes the distortion in the vectors during dimensionality reduction whereas the objective should be to minimize distortion during linear regression.



(a) Covariance of w_s vectors for spectral envelope (MGC: see Section V-A)



(b) Covariance of w_s vectors for fundamental frequency (LF0: see Section V-A)

Fig. 1. Covariances of weight vectors (w_s) for spectral envelope (MGC) and fundamental frequency (LF0) extracted from 88 speakers using 10 utterances per speaker using Eq. 9 are shown. Covariances of the 2, 5, and 10 dimensional weight vectors are shown separately. For an R-dimensional case, an $R \times R$ image is plot where intensity of each pixel is determined by the magnitude of the corresponding element in the covariance matrix.

The partial least squares (PLS) linear regression approach is used here to solve the generalization problem. In this approach, the input weight vector is

$$\boldsymbol{w}_{\text{s,in}} = \boldsymbol{\Gamma}\boldsymbol{x}_{\text{s,in}} + \boldsymbol{\epsilon}_{\text{s,in}} \tag{17}$$

and the output weight vector is

$$\boldsymbol{w}_{\text{s,out}} = \boldsymbol{\Omega}\boldsymbol{x}_{\text{s,in}} + \boldsymbol{\epsilon}_{\text{s,out}} \tag{18}$$

where the regression matrices $\boldsymbol{\Gamma} \in \mathbb{R}^{R \times R_r}$ and $\boldsymbol{\Omega} \in \mathbb{R}^{R \times R_r}$.

Because $R_r < R$, the dimensionality of the latent $\boldsymbol{x}_{\text{s,in}}$ vectors is lower than the dimensionality of $\boldsymbol{w}_{\text{s,in}}$ vectors. Thus, dimensionality of $\boldsymbol{w}_{\text{s,in}}$ is reduced in the first equation and a linear regression function is defined between the $\boldsymbol{x}_{\text{s,in}}$ and $\boldsymbol{w}_{\text{s,out}}$ vectors in the second equation. Combining those two equations, the linear regression function becomes

$$\boldsymbol{w}_{\text{s,out}} = \boldsymbol{\Psi}\boldsymbol{w}_{\text{s,in}} + \boldsymbol{\epsilon}_{\text{s}} \tag{19}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{R \times R}$. The solution with PLS minimizes $\sum_s \|\boldsymbol{\epsilon}_{\text{s}}\|^2$. The SIMPLS algorithm is used to solve the PLS regression problem [25].

3) Recursive Weighted Partial Least-squares Regression (rPLS): Some of the predictor variables in $\boldsymbol{w}_{s,in}$ are probably more important than others for explaining the observed variables in $\boldsymbol{w}_{s,out}$ through linear regression. One way to handle that in PLS is to use a method such as jack-knife [26] and remove unimportant variables. However, assigning weights to variables depending on their prediction power can lead to a more accurate solution. Recursive PLS (rPLS) algorithm is used here to perform such importance weighting [27].

If the vectors $\boldsymbol{w}_{\text{s,out}}$ and $\boldsymbol{w}_{\text{in}}$ are preprocessed to have zero mean and unit variance, then for each element i of $\boldsymbol{w}_{\text{s,out}}$, $\boldsymbol{w}_{\text{s,out}}(i)$, PLS algorithm can be used independently so that

$$\boldsymbol{w}_{\text{s,out}}(i) = \boldsymbol{b}_i\boldsymbol{w}_{\text{s,in}}, \tag{20}$$

where $\boldsymbol{b}_i$ is the regression vector for estimating $\boldsymbol{w}_{\text{s,out}}(i)$. After a PLS solution is found, $\boldsymbol{b}_i$ can be used for importance weighting. In that case, the input vectors from the previous iteration are reweighted using

$$\boldsymbol{w}_{\text{s,in}}^{iter} = \boldsymbol{w}_{\text{s,in}}^{iter-1} diag(\boldsymbol{b_i}). \tag{21}$$

where diag($\boldsymbol{b}_i$) is a diagonal matrix where the elements of $\boldsymbol{b}_i$ are on the diagonal. PLS is then used again to re-estimate b_i. The PLS and weighting steps are iterated until convergence.

Note that rPLS performs importance weighting for each element of $\boldsymbol{w}_{\text{s,out}}$ independently. Thus, the rPLS model is trained independently for each element of $\boldsymbol{w}_{\text{s,out}}$ which could cause degradation if there is high correlation between the elements of $\boldsymbol{w}_{\text{s,out}}$.

4) Weighted Partial Least-squares Regression (WPLS): Similar to weighted linear regression, weighted PLS (WPLS) can be used for weighting the eigenvoice vectors depending on their importance, during training. In this approach, the eigenvectors of the training speakers can be weighted such that $\sum_{s=1}^{N_p} \boldsymbol{w}_s||\boldsymbol{\epsilon}_s||^2$ is minimized, where $\boldsymbol{w}_s$ is the weight for speaker s. In our case, the weights are proportional to the normalized distances of target speakers to training speakers and they can be incorporated into the PLS training algorithm simply by duplicating the training samples in proportion to their weight as described below.

Let the weight of each training speaker i be equal to $L_{tar}(i)$ defined in Eq (16). Then, the data for each training speaker i can be repeated $r_i = round(N_r \times w_i)$ times in the training set where N_r is an integer constant. Those repetitions will approximately increase the size of the training database by a factor of N_r. If the SIMPLS training algorithm is used, the contribution of each sample to the total error $\boldsymbol{\epsilon}$ will be equally weighted. However, because each sample is repeated r_i times and same error $\boldsymbol{\epsilon}_i(r)$ is obtained for each repetition r, total error contributed by speaker i, $\boldsymbol{\epsilon}_i$, is equal to $r_i||\boldsymbol{\epsilon}_i(r)||^2$ where r_i is proportional to w_i if we ignore the round-off effects. Thus, minimization of the total error with the SIMPLS algorithm will minimize weighted errors when samples are duplicated in proportion to their weights.

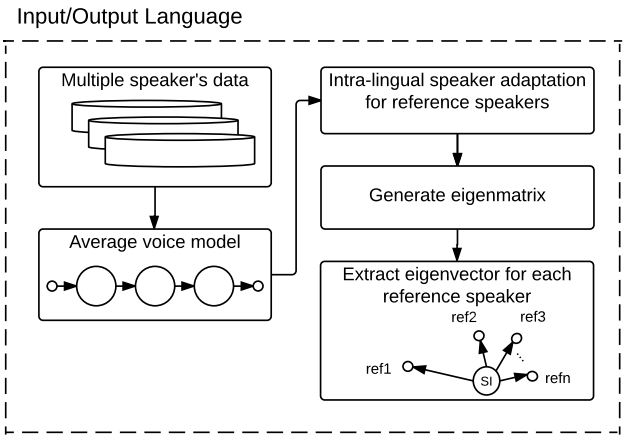


Fig. 2. Generation of the eigenspace and extraction of weight vectors for reference speakers. The procedure is done for both input and output languages while performing cross-lingual adaptation using eigenvector mapping.

Because the approach proposed here does not change the training algorithm – it only modifies the training dataset – it can also be used with rPLS, giving us weighted rPLS (WRPLS). Steps for training the AVMs and extracting the eigenvector for each reference speaker in input or output languages are shown in Figure 2. An overview of the various eigenvoice mapping cross-lingual adaptation algorithms is shown in Figure 3.

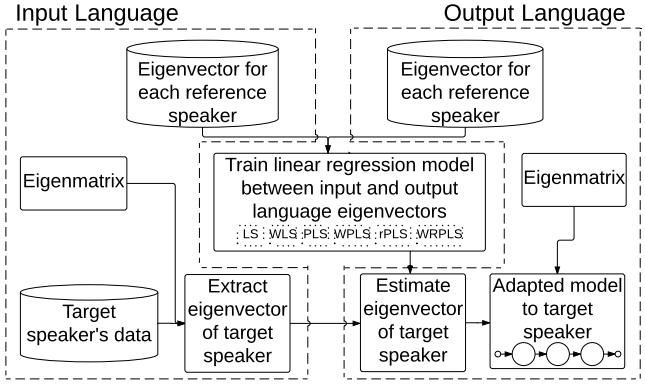


Fig. 3. Cross-lingual adaptation of a target speaker to an output language using eigenvector mapping.

B. Algorithms based on data-mapping

1) Nearest-neighbour state-mapping: The baseline algorithm performs state-mapping using the AVMs once and uses the same map for all target speakers. However, data mapping could be more effective if the state-mapping were done in a speaker-specific manner. To that end, separate speaker-dependent models of each reference speaker were adapted for each of the input and output languages.

A cross-lingual state map was learned separately for each of those training speakers, using their speaker-dependent models. As a result, for each bilingual training speaker s_i, a map M_i between that speaker's models for the input and output languages was produced.

Our proposal is to select one of those pre-trained maps to use for adaptation of a (previously unseen) target speaker. Similarity between the target speaker and the training speakers can be used to select the nearest training speaker, S_{nn}, to the target speaker s_{tar}. Euclidean distance, $(\boldsymbol{\mu}_{nn}-\boldsymbol{\mu}_{tar})^T(\boldsymbol{\mu}_{nn}-\boldsymbol{\mu}_{tar})$, is used as the similarity measure, where $\boldsymbol{\mu}_{nn}$ is the supervector of state means in the input language model of nearest training speaker. Similarly, $\boldsymbol{\mu}_{tar}$ is the supervector of the target speaker.

Once S_{nn} is selected, the state-map $\boldsymbol{M}_{nn}$ is used for mapping the adaptation data to output language states. Then, similar to the baseline approach, intra-lingual adaptation is performed.

2) Eigenvoice adaptation using data-mapping: Cross-lingual Bayesian eigenvoice adaptation (Cross-BEA) can be performed using a data-mapping approach once a state-map $\boldsymbol{M}_{tar}$ is available for the target speaker. Here, the nearest-neighbour based state-mapping algorithm described above is used to find $\boldsymbol{M}_{tar}$.

Once the adaptation data is mapped to the states of the output language, computation of $\hat{\boldsymbol{w}}_{s,out}$ is exactly the same as the intra-lingual adaptation case. The adaptation data-dependent variables $\boldsymbol{S}_{x,c}^{(s)}$ and $N_c^{(s)}$ in Eq (6) are computed by mapping data to output language states using $\boldsymbol{M}_{tar}$. Then, $\boldsymbol{w}_{s,out}$ is estimated using Eq (3). Steps for finding the nearest reference to the target speaker is shown in Figure 4. A diagrammatic overview of all algorithms based on data mapping is in Figure 5.

V. Experiments

A. Experimental settings

All systems in our experiments employed 78 dimensional observation vectors comprising 24 Mel-Generalized Cepstral Coefficients (MGCs), 1 log-energy, 1 log-F0 (LF0) coefficient, and their delta and delta-delta parameters. A 25 msec analysis window with 5 msec frame shift is used for feature extraction. Phonemes are modelled with 5 state Hidden Semi-Markov Models (HSMM).

Turkish is the input language and English is the output language. Two male (bdl and rms) and two female (slt and clb) speakers from the CMU-ARCTIC database (1130 utterances per speaker) were used to train the average voice model (AVM) for English. For training the AVM in Turkish, speech from three female speakers (1100 utterances each) were used. For the purposes of testing the proposed methods, a bilingual

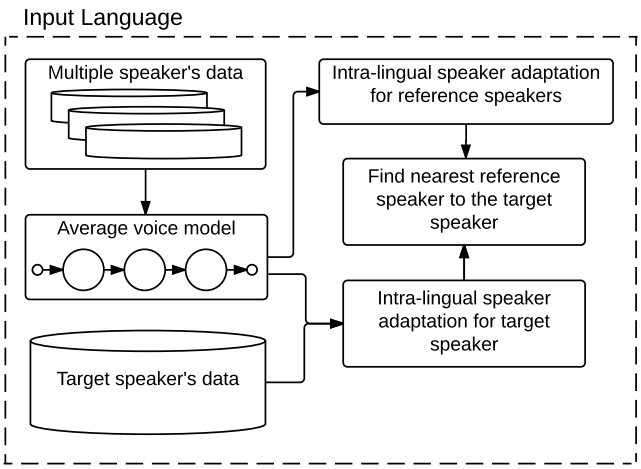


Fig. 4. Overview of the algorithm for finding the nearest reference speaker to the target speaker in input language.

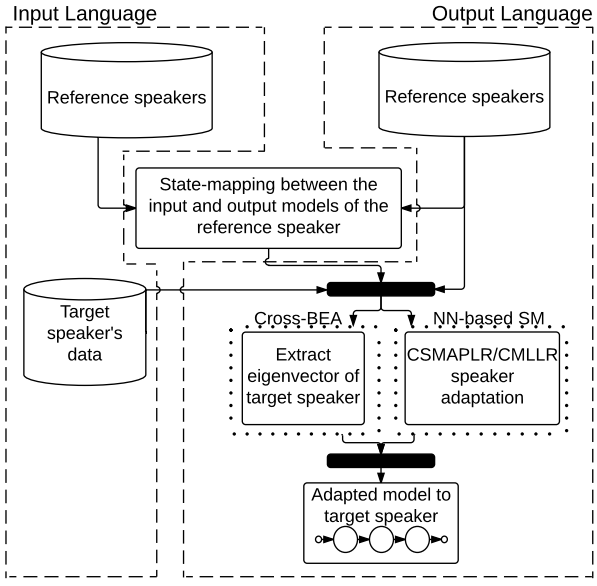


Fig. 5. Overview of the data-mapping algorithms. After state-mapping, proposed eigenvoice adaptation or CSMAPLR/CMLLR adaptations can be done. Those two options are shown between horizontal fork/join bars.

Turkish-English database was created, containing speech from 88 female speakers. From 29 speakers, used as targets, 50 Turkish and 50 English utterances were recorded. 10 Turkish and 10 English utterances were recorded by each of the remaining speakers. For better comparison between reference speakers, same sentences were used for all speakers.

For each speaker, a Turkish speaker-dependent model was created using the Turkish AVM and CSMAPLR adaptation followed by MAP adaptation. Similarly, English speaker-dependent models were created using the English AVM for each speaker. A leave-one-out method was used in testing for each of the 29 training speakers in turn. Thus, 87 training speakers were used for each target speaker. The rank hyper-parameter of the PLS and rPLS algorithms was tuned using cross-validation. The N_r parameter of the WRPLS algorithm was empirically set to 100.

TABLE I
α VALUES USED FOR 2, 5, OR 10 UTTERANCES OF ADAPTATION DATA, FOR ENGLISH AND TURKISH.

	English			Turkish		
	2 utt	5 utt	10 utt	2 utt	5 utt	10 utt
MGC	100	100	100	2000	10000	10000
LF0	25	100	100	500	1000	2000

The state-mapping algorithm [5] described in Section II was used as the comparison baseline since in similarity case, it is one of the best performing cross-lingual adaptation techniques available [8], [11].

Performance was measured with both objective and subjective tests. The objective test results are presented in Section V-B and the subjective test results are presented in Section V-C. The first set of objective tests were done to tune the regularization parameter, α, of the eigenvoice adaptation technique discussed in Section III. Then, the objective test results of the proposed data-mapping based algorithms are presented in Section V-B2. Performance of the eigenvector-mapping methods LS, WLS, PLS, and WPLS are discussed in Section V-B3 and the rPLS algorithm is discussed in Section V-B4. Finally, the best performing methods are compared in Section V-B5 and the most important findings are summarized in Section V-B6.

The subjective test results are presented for the best performing algorithms in Section V-C. Speaker similarity test results are discussed in Section V-C1 and the speech quality test results are discussed in Section V-C2.

B. Objective Measures

Root-mean-square-error (RMSE) is used for objectively measuring the distortion in LF0 features, with respect to natural references. Similarly, Mel-cepstral distortion (MCD) [28] is used for the MGC features. Synthetic speech from speaker-dependent models was played to listeners as the reference samples. The duration model of the English AVM was used in all cases [29] and so the duration of reference and test samples is always the same.

For each target speaker, adaptation was performed using 2, 5, or 10 utterances of adaptation data. For each adapted model, 40 English sentences from the WSJ1 database were synthesized for testing. Significance of the difference between models was measured with a t-test at 95% confidence interval.

1) Tuning the regularization parameter: The hyper-parameter α that is used in the regularized eigenvoice approach described in Section III was tuned experimentally for LF0 and MGC features. Tuning was done for Turkish and English voices separately as shown in Figure 6. The values of α used in the experiments are given in Table I.

When α increases, the possibility of overfitting decreases. However, if α is too high, then the algorithm does not have enough flexibility to adapt. For Turkish, regularization helped significantly both for LF0 and MGC features.

In the case of English, overfitting did not generally occur. Although a little overfitting occurred for the 10 PCA case, it

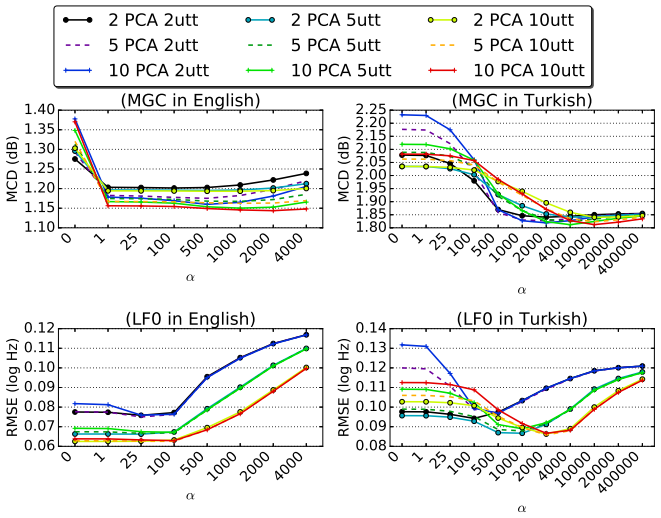


Fig. 6. Performance of regularization in intra-lingual adaptation for MGC and LF0 features in English and Turkish with different α values. Note that for the LF0 features, a difference of 0.01 log(Hz) corresponds to 17.3 cents.

was not significant for MGC and significant for LF0 only in the 2 or 5 adaptation utterance situations.

There are differences between Turkish and English that explain the differing behaviour regarding regularization. The target speakers are native speakers of Turkish and so their speech is well modelled by the average voice model and their prosodic and pronunciation patterns are consistent when they speak Turkish. For English, this is not the case. Therefore, stronger patterns and higher variability was observed in the case of Turkish, as shown in Figure 7 where the eigenvalues obtained for Turkish and English are shown.

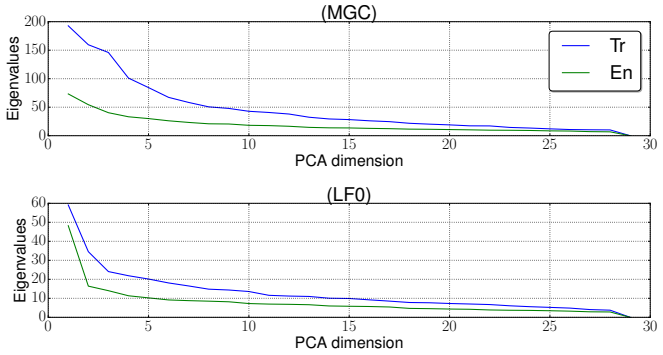


Fig. 7. Eigenvalues of reference speakers in Turkish and English languages.

2) Objective performance of algorithms based on data-mapping: Two algorithms proposed here are based on data-mapping (Section IV). Performance of the NN-based state-mapping algorithm is compared with the other algorithms in Figure 8. Because CSMAPLR and CMLLR can each be used after data is mapped to output language AVM states, both were tested in combination with the baseline and proposed state-mapping methods. The proposed NN-based state-mapping algorithm significantly outperformed the baseline algorithm both for MGC and LF0 and for all adaptation data sizes. CSMAPLR and CMLLR performed equally well for the MGC features. For LF0, CMLLR performed better than CSMAPLR for the baseline system, and CSMAPLR performed better for the proposed system.

The baseline and NN-based state-mapping algorithms were also compared with the Cross-BEA method in Figure 8 when 2-, 5-, and 10-dimensional eigenspaces were used. The Cross-BEA method substantially improved the performance compared to other techniques, for the MGC features.

For LF0, the Cross-BEA algorithm did not perform as well as NN-based state-mapping. Because the state-mapping accuracy is high when NNs are used, low dimensional LF0 vectors could be adapted well with CSMAPLR. However, performance of the eigenvoice algorithm saturated quickly and it so it does not perform as well as CSMAPLR as the amount of data grows: the performance gap widens with increasing data size.

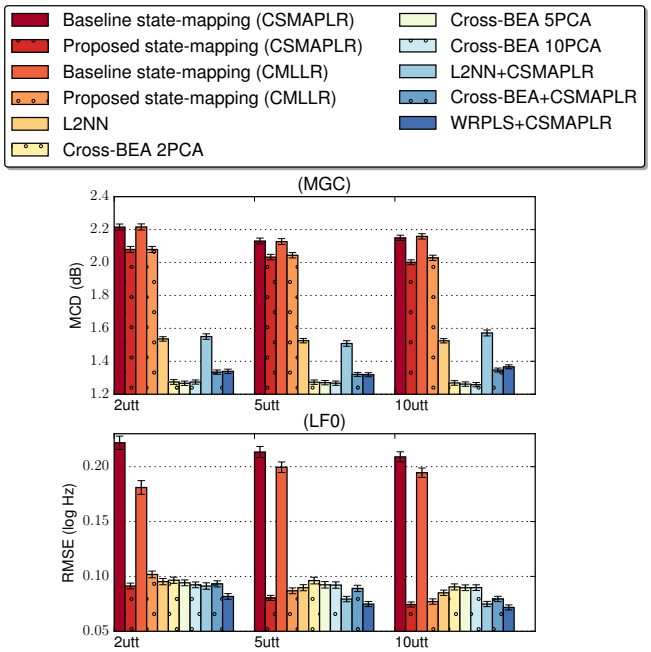


Fig. 8. Objective evaluation (RMSE and MCD) of algorithms based on data-mapping for MGC and LF0 features, showing 95% confidence intervals. The groups of results for "2utt", "5utt" and "10utt" correspond to 2, 5 and 10 utterances of adaptation data. Note that for the LF0 features, a difference of 0.1 log(Hz) corresponds to 173 cents. Cross-BEA+CSMAPLR was done with 10 dimensional PCA. WRPLS+CSMAPLR was done with 2 dimensional PCA.

Algorithms that use data mapping were also compared with the case where speech is synthesized with the nearest-neighbour (L2NN) without any further adaptation. Even though this approach worked well, as shown in Figure 8, it did not perform better than the Cross-BEA algorithm for the MGC features and the NN-based state-mapping algorithm with CSMAPLR for the LF0 features.

Additional CSMAPLR adaptation was done after L2NN, Cross-BEA, and WRPLS algorithms to investigate if there is opportunity for further improvement with additional adaptation steps. Results are shown in Figure 8. CSMAPLR degraded the performance for the high-dimensional MGC features when applied after L2NN, Cross-BEA, and WRPLS algorithms.

Thus, the CSMAPLR algorithm overfit on the adaptation data for the high-dimensional MGC features and that distorted the models. However, it helped improve the performance for the low-dimensional LF0 features.

3) Objective performance of least-squares algorithms: Performance of the LS, WLS, PLS, and WPLS algorithms for the MGC and LF0 features is shown in Figure 9.

MGC Features: For the 2 utterance case, the differences between the algorithms are not significant. For the 5 utterance case, WLS performed significantly better than LS for all PCA sizes but WPLS is not significantly better than PLS. In the 10 utterance case, for 2 and 5 dimensional PCA, all four algorithms performed equally well. For the 10 dimensional PCA case, PLS and WPLS substantially outperformed the LS and WLS algorithms. This is expected, since the variances of the eigenvectors increase with more data and it becomes harder to predict the English eigenvectors using linear regression. By exploiting correlations between eigenvector elements, PLS is able to do the regression in a lower dimensional space and avoid overfitting.

Note that objectively-measured performance of linear regression algorithms generally becomes worse with increasing data: models deviate further from the AVM. The small number of training speakers and non-linear relationship between input and output eigenvectors cause degradation. Thus, for the MGC features, performance with 2 utterances is actually better than with 5 or 10 utterances.

LF0 Features: Weighting the samples did not generally have a significant effect on performance in the 2 utterance case (except for 5-dimensional PCA), as shown in Figure 9. For 5-dimensional PCA, partial least-squares (PLS, WPLS) is worse than straightforward least-squares (LS, WLS).

For the 5 utterance case, all algorithms performed equally well, except that LS and WLS were significantly worse than the others for 10-dimensional PCA. Similarly to the situation for MGC features, the partial least-squares (PLS) algorithm solved the overfitting exhibited by least-squares (LS, WLS) for the 5 utterance, 10-dimensional PCA case.

In the 10 utterance, 2-dimensional PCA case, least-squares (LS, WLS) outperformed partial least-squares (PLS, WPLS); this is as expected, because the correlations between the elements of the eigenvoice vectors are minimal for the 2-dimensional case, as we saw in Figure 1.

In contrast to MGC features, linear regression for LF0 performed better with more data. That is, the linear regression approach performs better for lower dimensional feature vectors. Moreover, degradation of performance with higher PCA sizes did not occur for LF0, except for the 5 utterance, 10-dimensional PCA case; this can be solved with PLS or WPLS.

4) Objective performance of the rPLS algorithm: The results above show that weighting the samples sometimes improves (and never reduces) the performance of LS and PLS. Hence, the remaining objective evaluations are presented for weighted least-squares (WLS, WPLS) only. In Figure 10, performance of the weighted least-squares (WLS, WPLS) algorithms is compared with the recursive variants (rPLS, WRPLS). Although rPLS did not perform well (at most PCA sizes for the 5 utterance and 10 utterance cases, for MGC features), WRPLS was consistently the best performing algorithm for all amounts of data and at all PCA dimensions. This indicates that weighting is effective and should be speaker-specific.

Note that the rPLS algorithm works independently for each element of the eigenvector in the output language. This means that any correlations between elements of the vector violate the independence assumption and are therefore likely to degrade performance. Covariance matrices for the MGC features is shown in Figure 1: even though the matrix for 2-dimensional PCA is diagonal, substantial covariances can be observed for the 5- and 10-dimensional cases; this explains the relatively poor performance of rPLS for MGC features (Figure 10.)

For LF0, no particular algorithms consistently and significantly outperforms the others. This is probably because of relatively weak correlations between the elements of LF0 eigenvectors (cf. Figure 1).

5) Direct comparison of the best performing algorithms: WRPLS, which is the best performing linear regression based algorithm, is now compared with the best performing data-mapping algorithm, Cross-BEA. Figure 11 shows the performance of these approaches across different amounts of data and different PCA dimensions. The performance of intra-lingual adaptation is included in the figure, as an upper bound.

For the MGC features, WRPLS outperforms Cross-BEA when only 2 utterances are available; the two algorithms become comparable with 5 utterances, and Cross-BEA outperforms WRPLS algorithm (at all PCA dimensions) when there are 10 utterances. The performance gap between the algorithms increases with PCA dimension.

The situation is reversed for LF0 features. With only 2 utterances, Cross-BEA performs better than WRPLS (at all PCA dimensions). With 5 utterances, WRPLS and Cross-BEA perform similarly, then WRPLS slightly outperforms Cross-BEA when there are 10 utterances.

NN-based state-mapping with CSMAPLR was also compared with WRPLS and Cross-BEA for the LF0 feature and it outperformed them both substantially in the 5 and 10 utterances cases. For those relatively larger data sizes, even though a more accurate state mapping is available, Cross-BEA is not able to exploit the data effectively because its performance has already saturated. In contrast, CSMAPLR performance keeps improving with increasing data (for the LF0 features). This also partly explains why WRPLS outperforms Cross-BEA with increasing data size.

6) Summary of objective performance: A large number of objective comparison tests have been presented above. The most important findings are:

- NN-based state-mapping outperforms baseline state-mapping for both MGC and LF0 features. This is shown by objective experiments presented in Figure 8. Thus, using speaker-dependent state-mapping was found to be effective compared to speaker-independent state-mapping.
- Cross-BEA performs substantially better than the CSMAPLR algorithm for the MGC features as shown in Figure 8. Hence, the CSMAPLR algorithm could not adapt the high-dimensional MGC features as well as the eigenvoice adaptation algorithm with the limited data.

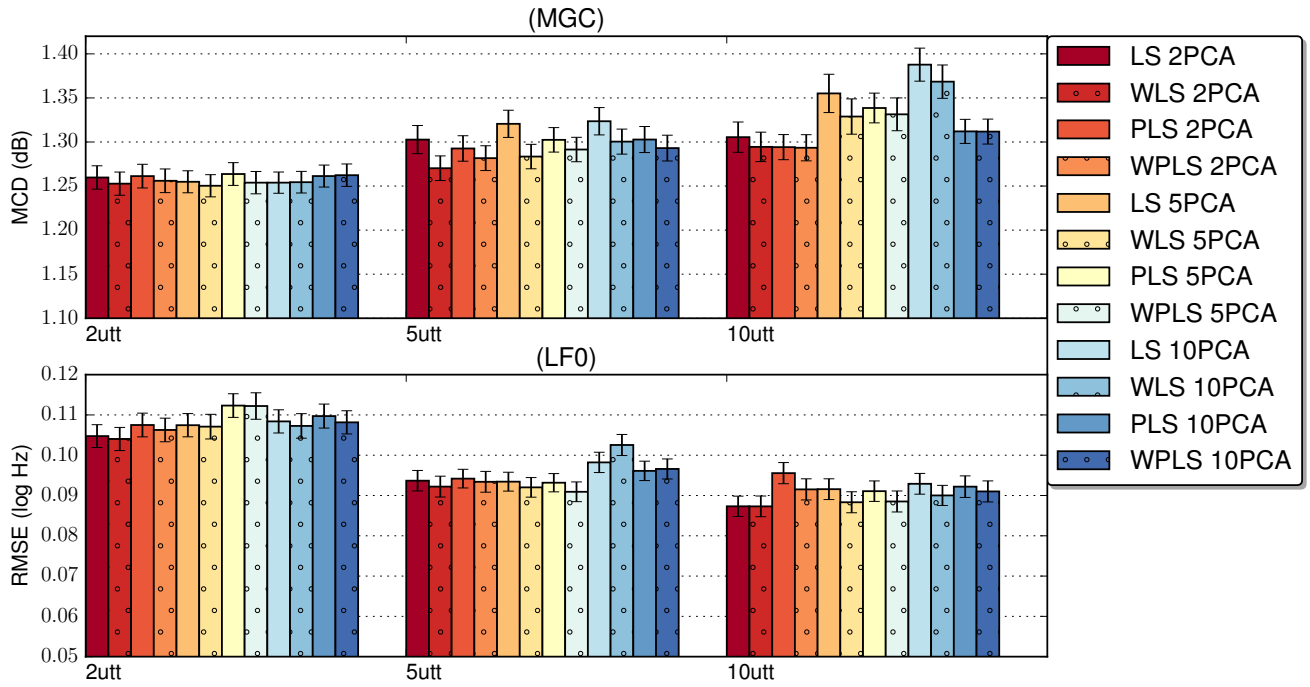


Fig. 9. Objective evaluation (RMSE and MCD) of the LS, WLS, PLS, and WPLS algorithms for MGC and LF0 features using 2, 5 and 10 dimensional PCA with 95% confidence intervals. The plots for "2utt", "5utt" and "10utt" correspond to 2, 5 and 10 utterances of adaptation data. Note that for the LF0 features, a difference of 0.01 log(Hz) corresponds to 17.3 cents.

- Using the nearest-neighbour model without any further adaptation performed significantly better than the baseline system as shown in the Figure 8. This indicates that a nearest-neighbour model trained with intra-lingual adaptation is preferable to a model trained with the baseline algorithm using limited data if the nearest-neighbour sounds similar to the target speaker.
- For the MGC features, eigenvector mapping becomes relatively less effective with increasing adaptation data. Importance weighting and PLS regression improved the performance, although combining them together did not further improve performance as shown in Figure 9 and Figure 10. PLS approach helped reduce the overfit problem because it does regression in a lower dimensional space. Importance weighting addresses the non-linear relationship between the input and output vectors during regression by assuming piecewise linearity. One reason the combination of the two did not further improve the performance could be because of a reduction in non-linearity in the lower dimensional space that the PLS regression operates in.
- For LF0, eigenvector mapping becomes more effective with increasing adaptation data size. Because the feature dimensionality is much lower for LF0, even the basic least-squares (LS) approach performs well, regardless of the amount of adaptation data as shown in Figure 9 and Figure 10.
- rPLS did not perform well, presumably because of correlations in the features. However, weighting remedied this substantially and WRPLS was the best performing algorithm for MGC and LF0, along with WLS as shown in Figure 10.
- Performance degrades significantly with increasing PCA size for all regression algorithms, especially with 5 or 10 utterances, due to overfitting and non-linearities; the issue is more significant for MGC features as shown in Figure 9 and Figure 10.
- For the MGC features, Cross-BEA performs better than the best performing regression method, WRPLS, with the largest amount of adaptation data (10 utterance). WRPLS performs better when only 2 adaptation utterances are available. The converse is true for LF0 as shown in Figure 11. The LF0 features are in a far smaller space compared to the MGC features and 2 utterances are enough for an effective Cross-BEA adaptation whereas larger amount of data is needed for the MGC features. WRPLS performs better than Cross-BEA for the LF0 features with larger data possibly because the relationship between the input and the output eigenvectors is more linear compared to the MGC case.
- NN-based state-mapping with CSMAPLR substantially outperforms both WRPLS and Cross-BEA for LF0 as shown in Figure 8 and Figure 11. Thus, CSMAPLR can do effective adaptation for the low-dimensional LF0 features with limited amounts of data and eigenvoice based techniques are not necessary if CSMAPLR is used with the NN-based approach.

C. Subjective evaluation

1) Speaker similarity tests: To subjectively measure the similarity of the adapted speaker to the target speaker we

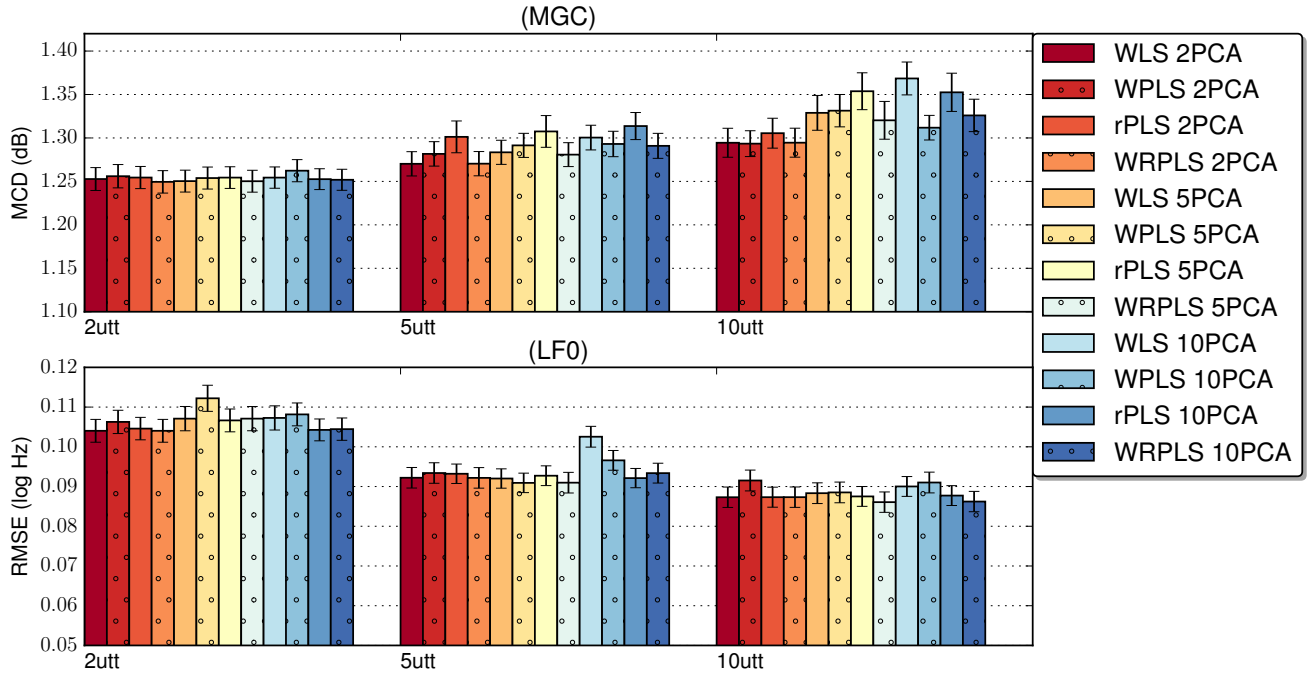


Fig. 10. Objective performance (RMSE for LF0 and MCD for MGC features) of the WLS, WPLS, rPLS, and WRPLS algorithms using 2, 5 or 10 dimensional PCA; 95% confidence intervals are shown. Note that for the LF0 features, a difference of 0.01 log(Hz) corresponds to 17.3 cents.

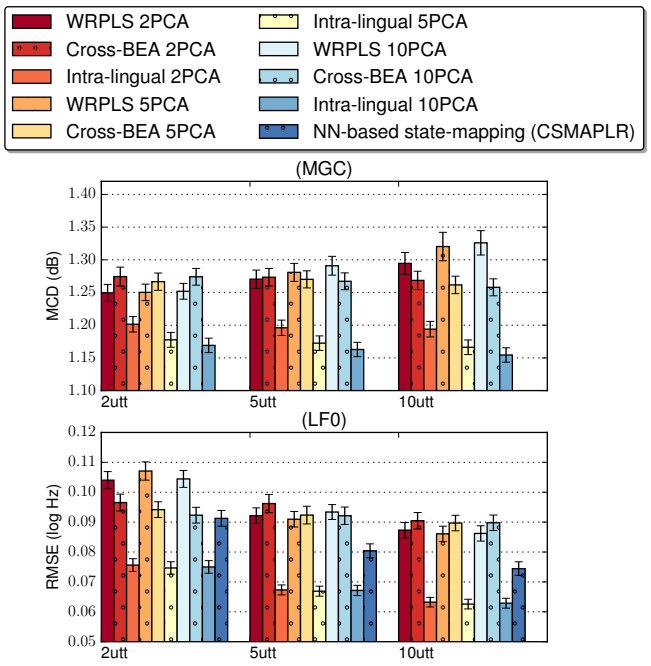


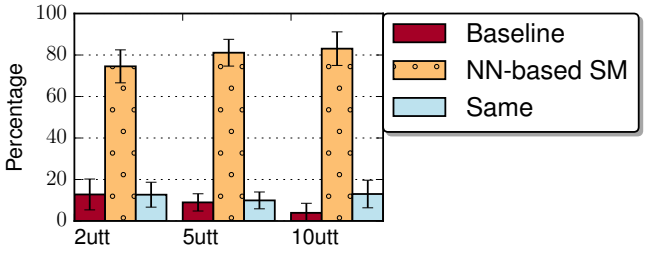
Fig. 11. Objective evaluation (RMSE of LF0 and MCD of MGC features) of the best performing algorithms WRPLS and Cross-BEA. NN-based state-mapping is shown for LF0 only. 95% confidence intervals are shown. Intra-lingual adaptation performance is included as an upper-bound. Note that for the LF0 features, a difference of 0.01 log(Hz) corresponds to 17.3 cents.

employed ABX testing. As with the objective measures, synthetic speech from speaker-dependent models was used as the reference X. Listeners were asked to select which of the speakers of sample A or sample B was more similar to this, or to indicate that samples A and B sounded the same in terms of similarity to X. The A and B samples were synthesized from different adaptation methods randomly. 10 target speakers were selected randomly and, for each speaker, five English sentences from the WSJ1 database were synthesized for each amount of adaptation data (2, 5, or 10 utterances). The tests were done in two phases. In the first phase, 12 native (10 female and 2 male) listeners and 2 non-native male listeners took the tests in soundproof booths and they all listened to one utterance from each speaker. Even though those utterances were different for different speakers, they were the same for all listeners given a speaker. In the second phase, a different set of 12 gender-balanced native English speakers took the tests. In this phase, each listener judged one utterance from each speaker and the utterances were randomly selected out of four utterances synthesized for each speaker. Results from the two phases are combined for analysis.

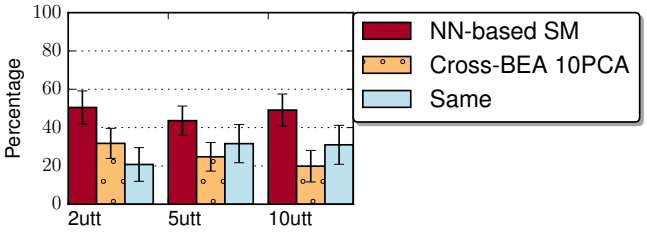
The average age of listeners was 22 years. The stimuli were presented over headphones and listener responses were collected via a simple web browser interface. Listeners could play the A, B and X samples as many times as they desired and they were informed about that before the test. However, they were not encouraged or discouraged to do that. In each test, 30 samples were played to each listener and in average it took 15 minutes to finish the test. The text was the same in A, B and X within a single presentation.

Guided by the objective results, four subjective ABX tests were designed. In the first, the performance of the baseline state-mapping algorithm, generic state-mapping with no information from the target speaker, with CMLLR for LF0 was compared with the proposed NN-based state-mapping algorithm with CSMAPLR; this was the best performing algorithm

for LF0 according to objective measures. In both cases, Cross-BEA (10-dimensional PCA) was used to generate the MGC features. The results are shown in Figure 12a. Clearly, the NN-based state-mapping algorithm substantially outperforms the baseline state-mapping algorithm (which uses the same state-map for all speakers).

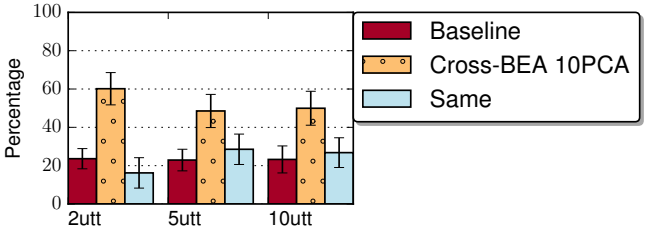


(a) LF0 generated with either the baseline vs. the NN-based state-mapping (SM).

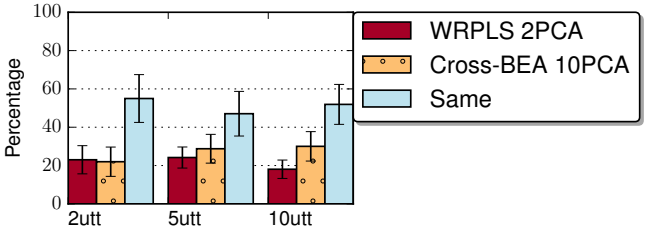


(b) LF0 generated using the NN-based state mapping vs. the Cross-BEA with 10 dimensional PCA.

Fig. 12. Listeners' preferences for the speaker similarity of synthetic speech in which LF0 was generated using different adaptation algorithms. 95% confidence intervals are shown. 2, 5, or 10 utterances were used for adaptation.



(a) Baseline vs. Cross-BEA with 10-dimensional PCA.



(b) WRPLS with 2-dimensional PCA vs. Cross-BEA with 10-dimensional PCA.

Fig. 13. Listeners' preferences for the speaker similarity of synthetic speech in which MGC features were generated using different adaptation algorithms. 95% confidence intervals are shown. 2, 5, or 10 utterances were used for adaptation.

In the second experiment, the proposed NN-based state-mapping algorithm with CSMAPLR for LF0 was compared with Cross-BEA (10-dimensional PCA). As before, Cross-BEA (10-dimensional PCA) was used to generate the MGC features. Results are shown in Figure 12b. Even though the gap is not as dramatic as in the first experiment, we see that the proposed NN-based state-mapping approach significantly outperformed the Cross-BEA algorithm, for all adaptation data amounts.

In the third experiment, MGC features generated using the baseline state-mapping algorithm with CSMAPLR were compared with those from Cross-BEA (10-dimensional PCA), which was the best performing algorithm for the MGC features according to the objective measure (MCD). The NN-based state mapping algorithm was used to generate LF0 in both cases. Results are shown in Figure 13a where we can see that Cross-BEA is substantially preferred over the baseline system.

In the final ABX experiment, the WRPLS algorithm was compared with Cross-BEA (10-dimensional PCA) for MGC features. Again, the NN-based state mapping algorithm was used to generate F0. Results are shown in Figure 13b which reveals that listeners had no particular preference for WRPLS or Cross-BEA.

2) Speech quality tests: For evaluation of the speech quality with the proposed methods, the MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) test was conducted. The samples were synthesized using the models generated with the best performing proposed adaptation methods and the baseline method in a random order. Five target speakers were selected randomly and, for each speaker, five English sentences from the WSJ1 database were synthesized with the models adapted with 2 and 10 utterances.

14 native (7 female and 7 male) listeners took the tests in soundproof booths. The average age of listeners was 24 years. The test is composed of 25 sets where each set contains 9 stimuli of the same sentence generated by each of the four adaptation systems (baseline, NN-base state-mapping with CSMAPLR, WRPLS, and Cross-BEA) for the 2 and 10 utterance adaptation data cases. Synthetic speech from speaker-dependent models was used as the hidden reference. The listeners were asked to rate each stimulus from 0 (extremely bad in naturalness aspect) to 100 (same as natural speech).

The MUSHRA test results are presented in Figure 14. Paired t-test was used to assess the significance of difference between the systems. For adaptation with 2 utterances, all proposed methods performed significantly better than the baseline system. However, the proposed methods were not found to be significantly different from each other. For adaptation with 10 utterances, the differences between the baseline system, WRPLS 2PCA and Cross-BEA 10PCA methods were not significant but the NN-based state-mapping method performed significantly better than them. Increasing the adaptation data size improved the performances of the baseline and the NN-based state mapping methods. But it does not have a significant effect on the WRPLS 2PCA and the Cross-BEA 10PCA methods.

VI. Conclusion and Future Work

We have investigated a variety of cross-lingual speaker adaptation algorithms for HMM-based speech synthesis sys-

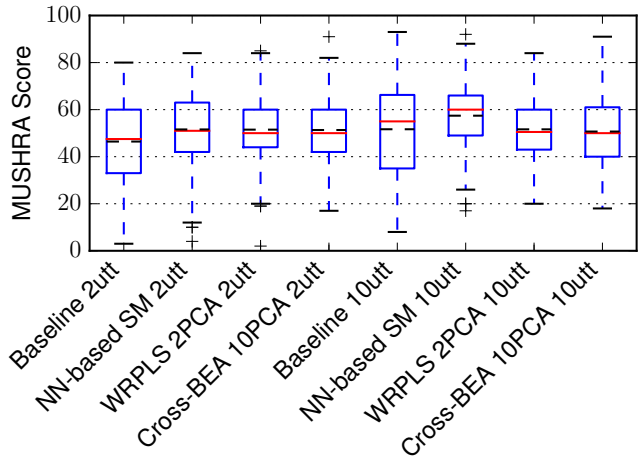


Fig. 14. Box plot of MUSHRA result for quality evaluation of the best performing algorithms. The bottom and top of each box are the first and third quartiles, respectively. Ends of the whiskers represent 1.5*IQR (InterQuartile Range)* distances from the first and third quartiles. Outliers are shown with "+" character. Median and mean of each box are shown with solid and dashed lines, respectively.

tems, with the specific use case of small amounts of adaptation data from the target speaker. This scenario is motivated by practical applications, in which users are unlikely to be patient enough to provide many minutes or hours of their speech.

We proposed two approaches, and compared them objectively and subjectively, using a Turkish-English bilingual voice database. In the first proposed approach, a speaker-specific state-mapping is constructed in which the state-map belonging to the nearest-neighbour (NN) speaker to the target speaker is used for adaptation. In the second proposed approach, linear regression is used to relate the eigenvectors of the input and output language acoustic models.

Both approaches performed better than the baseline state-mapping method, objectively and subjectively. The NN-based state mapping using CSMAPLR adaptation performed the best for LF0. The cross-lingual eigenvoice adaptation technique Cross-BEA performed the best for the MGC feature.

Eigenvoice spaces are trained independently in this work. A unified space for the input and output languages will be investigated in future work to improve the performance of linear regression between the eigenvectors. To that end, co-training those eigenspaces to produce linearly-dependent eigenvectors for the same speaker in the input and output languages will also be investigated.

Even though the algorithms that are proposed here are language-independent, experimenting with them for other language pairs is also interesting and will be investigated in future work. Cross-lingual adaptation between languages that are acoustically more similar to each other than the Turkish-English pair, Spanish and French or Turkic languages for example, will be the focus of our future work.

REFERENCES

[1] Shigeki Matsuda, Xinhui Hu, Yoshinori Shiga, Hideki Kashioka, Chiori Hori, Keiji Yasuda, Hideo Okuma, Masao Uchiyama, Eiichiro Sumita, Hisashi Kawai, et al., "Multilingual speech-to-speech translation system: VoiceTra," in *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*. IEEE, 2013, vol. 2, pp. 229–233.

[2] Keiichiro Oura, Junichi Yamagishi, Mirjam Wester, Simon King, and Keiichi Tokuda, "Analysis of unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis using KLD-based transform mapping," *Speech Communication*, vol. 54, no. 6, pp. 703–714, 2012.

[3] Hui Liang, Yao Qian, Frank K Soong, and Gongshen Liu, "A cross-language state mapping approach to bilingual (Mandarin-English) TTS," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4641–4644.

[4] Yi-Ning Chen, Yang Jiao, Yao Qian, and Frank K Soong, "State mapping for cross-language speaker adaptation in TTS," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4273–4276.

[5] Yi-Jian Wu, Yoshihiko Nankaku, and Keiichi Tokuda, "State mapping based method for cross-lingual speaker adaptation in HMM-based speech synthesis.," Interspeech, 2009, pp. 528–531.

[6] Hui Liang and John Dines, "An analysis of language mismatch in HMM state mapping-based cross-lingual speaker adaptation," in *Interspeech*, 2010, pp. 622–625.

[7] Xianglin Peng, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Cross-lingual speaker adaptation for HMM-based speech synthesis considering differences between language-dependent average voices," in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 605–608.

[8] Daiki Nagahama, Takashi Nose, Tomoki Koriyama, and Takao Kobayashi, "Transform mapping using shared decision tree context clustering for HMM-based cross-lingual speech synthesis," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[9] Javier Latorre, Koji Iwano, and Sadaoki Furui, "New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer," *Speech Communication*, vol. 48, no. 10, pp. 1227–1242, 2006.

[10] Heiga Zen, Norbert Braunschweiler, Sabine Buchholz, Mark JF Gales, Kate Knill, Sacha Krstulovic, and Javier Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1713–1724, 2012.

[11] Viviane de Franca Oliveira, Sayaka Shiota, Yoshihiko Nankaku, and Keiichi Tokuda, "Cross-lingual speaker adaptation for HMM-based speech synthesis based on perceptual characteristics and speaker interpolation," in *Interspeech*, 2012, pp. 983–986.

[12] Takenori Yoshimura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Cross-lingual speaker adaptation based on factor analysis using bilingual speech data for HMM-based speech synthesis," in *8th ISCA Speech Synthesis Workshop*, 2013, pp. 317–322.

[13] Malorie Charlier, Yamato Ohtani, Tomoki Toda, Alexis Moinet, and Thierry Dutoit, "Cross-language voice conversion based on eigenvoices," in *Interspeech*, 2009, pp. 1635–1638.

[14] Aanchan Mohan and Richard Rose, "Multi-lingual speech recognition with low-rank multi-task deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4994–4998.

[15] Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek, Tanja Schultz, and Hervé Bourlard, "Multilingual deep neural network based acoustic modeling for rapid language adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7639–7643.

[16] Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean, "Multilingual acoustic models using distributed deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8619–8623.

[17] Seyyed Saeed Sarfjoo and Cenk Demiroglu, "Cross-lingual speaker adaptation for statistical speech synthesis using limited data," *Interspeech*, pp. 317–321, 2016.

[18] Peng Liu, Frank K Soong, and Jian-Lai Thou, "Divergence-based similarity measure for spoken document retrieval," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 4, pp. IV–89.

[19] Vassilios V Digalakis, Dimitry Rtischev, and Leonardo G Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

[20] Mark JF Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[21] Junichi Yamagishi, Takao Kobayashi, Yuji Nakano, Katsumi Ogata, and Juri Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 66–83, 2009.

[22] Lakshmi Saheer, Hui Liang, John Dines, and Philip N Garner, "VTLN-based rapid cross-lingual adaptation for statistical parametric speech synthesis," Tech. Rep. Idiap-RR-12-2012, Idiap, 4 2012.

[23] Amir Mohammadi, Seyyed Saeed Sarfjoo, and Cenk Demiroglu, "Eigenvoice speaker adaptation with minimal data for statistical speech synthesis systems using a MAP approach and nearest-neighbors," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 2146–2157, 2014.

[24] K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Seventh International Conference on Spoken Language Processing*, 2002, pp. 1269–1272.

[25] Sijmen De Jong, "SIMPLS: an alternative approach to partial least squares regression," *Chemometrics and intelligent laboratory systems*, vol. 18, no. 3, pp. 251–263, 1993.

[26] Bradley Efron, *The jackknife, the bootstrap, and other resampling plans*, vol. 38, Society for Industrial and Applied Mathematics, Philadelphia, Pa, 1982.

[27] Åsmund Rinnan, Martin Andersson, Carsten Ridder, and Søren Balling Engelsen, "Recursive weighted partial least squares (rPLS): an efficient variable selection method using PLS," *Journal of Chemometrics*, vol. 28, no. 5, pp. 439–447, 2014.

[28] Robert F Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on*. IEEE, 1993, vol. 1, pp. 125–128.

[29] Yi-Jian Wu, Simon King, and Keiichi Tokuda, "Cross-lingual speaker adaptation for HMM-based speech synthesis," in *Chinese Spoken Language Processing, 2008. ISCSLP'08. 6th International Symposium on*. IEEE, 2008, pp. 1–4.