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

Citation for published version:

Digital Object Identifier (DOI):
10.1109/TASLP.2017.2667880

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published in:
IEEE/ACM Transactions on Audio, Speech, and Language Processing

Publisher Rights Statement:
(c) 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
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

‡†The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
*†Electrical and Computer Engineering Department, Ozyegin University, Istanbul, Turkey
*saed.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 regression further improved 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.
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 regression 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 $e_r \in \mathbb{R}^n$, which are called eigenvoices in this context, the mean supervector for speaker $s$ is $\mu(s) = \begin{bmatrix} \mu_1(s) & \mu_2(s) & \ldots & \mu_N_{st}(s) \end{bmatrix}^T$, where $N_{st}$ is the total number of states in the acoustic model, and $\mu_c(s)$ is the mean vector of the $c^{\text{th}}$ state; $\mu(s)$ can be modeled as:

$$\mu(s) = \mu_{sl} + E_w s + \epsilon_s$$

where $\mu_{sl}$ is the mean supervector of the AVM (i.e., a speaker-independent model), $E = [e_1, e_2, \ldots, e_R]$ is a matrix of eigenvectors spanning the space of speakers in the AVM, $w_s$ is the weight vector for speaker $s$, and $\epsilon_s$ is the approximation error.

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

$$p(X_s|w_s, E) \propto e^{\exp\left( -\frac{1}{2} \sum_{c=1}^{N_{st}} \sum_{i=1}^{N^{(c)}_s} (x^{(c)}_s - E_c w_s)^T \Sigma_c^{-1} (x^{(c)}_s - E_c w_s) \right)}$$

where $E_c \in \mathbb{R}^{F \times R}$ is the $c^{\text{th}}$ block of the $E$ matrix corresponding to state $c$, and $F$ is the size of the mean vectors.
\( x_c^{(i)} = x^{(i)} - \mu_c, \) is the \( i \)-th observation that is aligned with state \( c \). \( \mu_c \) and \( \Sigma_c \) are the speaker-independent mean vector and covariance matrix of the Gaussian emission pdf of state \( c \), and \( N_c^{(i)} \) 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 \( s \), \( w_s \in \mathbb{R}^{R \times 1} \), is estimated as
\[
\hat{w}_s = G^{(s)}_w^{-1} k^{(s)}_w
\]
where
\[
G^{(s)}_w = \sum_{c=1}^{N_{st}} N_c^{(s)} E_c^T \Sigma_c^{-1} E_c
\]
\[
k^{(s)}_w = \sum_{c=1}^{N_{st}} E_c^T \Sigma_c^{-1} S_{x,c}^{(s)}
\]

The weight vector of speaker \( s \), \( w_s \in \mathbb{R}^{R \times 1} \), is estimated as
\[
\hat{w}_s = G^{(s)}_w^{-1} k^{(s)}_w
\]
where
\[
G^{(s)}_w = \sum_{c=1}^{N_{st}} N_c^{(s)} E_c^T \Sigma_c^{-1} E_c
\]
\[
k^{(s)}_w = \sum_{c=1}^{N_{st}} E_c^T \Sigma_c^{-1} S_{x,c}^{(s)}
\]

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, \( p(w) \), on the weight vector. \( w_s \) 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
\[
\hat{w}_{s,\text{map}} = \argmax_w p(\chi_a|w)p(w)
\]
where \( p(w) \) is the prior, set to \( N(0, \Sigma_w) \) here.

Using Eq (2) to replace the likelihood term \( p(\chi_a|w_s) \), removing the terms that are independent of \( w \) from the objective function, and with some matrix manipulation, the MAP objective function becomes
\[
\hat{w}_{s,\text{map}} = \argmax_w \exp(w^T E^T \Sigma^{-1} S_x - 1/2 w^T E^T N \Sigma^{-1} E w) \exp(-1/2 w^T \Sigma_w^{-1} w),
\]

where the block diagonal \( \Sigma^{-1} = \text{diag}(\Sigma_1^{-1}, \Sigma_2^{-1}, ..., \Sigma_{N_{st}}^{-1}) \),
\[
S_x = [S_{x,1}, S_{x,2}, ..., S_{x,N_{st}}], \quad N = \text{diag}(N_1, N_2, ..., N_{N_{st}}).
\]

The objective function can be maximized by noting that the posterior distribution \( p(w|\chi_a) \) 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
\[
\hat{w}_{s,\text{map}} = \argmax_w \exp(-1/2 (w - \mu_{w|x})^T R_{w|x}(w - \mu_{w|x}))
\]
where \( R_{w|x} \) is the precision matrix. By completing the squares and using Eq (8),
\[
R_{w|x} = (E^T N \Sigma^{-1} E + \Sigma_w^{-1}),
\]
and
\[
\mu_{w|x} = R_{w|x}^{-1} E^T \Sigma^{-1} S_x.
\]

MAP estimate of \( w \), \( \hat{w}_{s,\text{map}} \), is the mean, \( \mu_{w|x} \), of the posterior distribution, \( \Sigma_w^{-1} \) is a hyper-parameter of the prior which we set to \( \alpha S^{-1} \) where \( \alpha \) is a scalar (chosen empirically) and \( S \) is the diagonal matrix
\[
S = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_R)
\]
where \( \lambda_i \) 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 adaptation: that is, they result in an estimate for \( w_{s,\text{in}} \). However, the weight vector for the output language, \( w_{s,\text{out}} \), is required for cross-lingual adaptation, so that we can compute
\[
\mu_{\text{out}}^{(s)} = \mu_{\text{out}}^{(s)} + E_{\text{out}} \hat{w}_{s,\text{out}}.
\]

where \( \mu_{\text{out}}^{(s)} \) is the supervector, \( E_{\text{out}} \) is the eigenvoice matrix, and \( \mu_{\text{out}}^{(s)} \) is the speaker-independent supervector for the output language. We have investigated both data-mapping and vector-/space-mapping techniques to estimate \( w_{s,\text{out}} \). Our proposed techniques are described below.

IV. CROSS-LINGUAL EIGENVOICE ADAPTATION

A. Algorithms based on eigenvector mapping

Given \( w_{s,\text{in}} \), computed using intra-lingual adaptation, we can use linear regression to predict \( w_{s,\text{out}} \). The \( w \) 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 \( A \) can be trained such that \( w_{s,\text{out}} = Aw_{s,\text{in}} + \epsilon \). In the simplest approach, the least-squares (LS) algorithm is used for training \( A \). 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 \( A \) 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 \( A_{\text{tar}} \) matrix is computed using:
\[
A_{\text{tar}} = \argmin_A \sum_{i=1}^{N_p} e_{i,\text{tar}}^T e_{i,\text{tar}}
\]
where \( N_p \) is the number of training speakers and
\[
e_{i,\text{tar}} = L_{\text{tar}}(i)(w_{\text{out}}(i) - Aw_{\text{in}}(i))
\]
where \( L_{\text{tar}}(i) \) is the weight of the \( i \)th training speaker, \( w_{\text{out}}(i) \) is its eigenvoice vector in the output language and \( w_{\text{in}}(i) \) is its eigenvoice vector in the input language.

The speaker weights \( L_{\text{tar}}(i) \) 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

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

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

Once \( L_{\text{tar}}(i) \) and \( A_{\text{tar}} \) are computed, the eigenvoice weight in the output language is estimated as \( w_{\text{tar.out}} = A_{\text{tar}} w_{\text{tar.in}} \).

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.

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

\[
w_{\text{s,in}} = \Gamma x_{\text{s,in}} + \epsilon_{\text{s,in}} \tag{17}
\]

and the output weight vector is

\[
w_{\text{s,out}} = \Omega x_{\text{s,in}} + \epsilon_{\text{s,out}} \tag{18}
\]

where the regression matrices \( \Gamma \in \mathbb{R}^{R \times R} \) and \( \Omega \in \mathbb{R}^{R \times R} \).

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

\[
w_{\text{s,out}} = \Psi w_{\text{s,in}} + \epsilon_s \tag{19}
\]

where \( \Psi \in \mathbb{R}^{R \times R} \). The solution with PLS minimizes \( \sum_s \| \epsilon_s \|^2 \). 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 \( w_{\text{s,in}} \) are probably more important than others for explaining the observed variables in \( w_{\text{s,out}} \) 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 \( w_{\text{s,out}} \) and \( w_{\text{in}} \) are preprocessed to have zero mean and unit variance, then for each element \( i \) of \( w_{\text{s,out}} \), \( w_{\text{s,out}}(i) \), PLS algorithm can be used independently so that

\[
w_{\text{s,out}}(i) = b_i w_{\text{s,in}}, \tag{20}
\]

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

\[
w_{\text{s,in}}^{\text{iter}} = w_{\text{s,in}}^{\text{iter}-1} \text{diag}(b_i). \tag{21}
\]

where \( \text{diag}(b_i) \) is a diagonal matrix where the elements of \( b_i \) 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 \( w_{\text{s,out}} \) independently. Thus, the rPLS model is trained independently for each element of \( w_{\text{s,out}} \) which could cause degradation if there is high correlation between the elements of \( w_{\text{s,out}} \).

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 \( \sum_{s=1}^{N_s} w_s \| \epsilon_s \|^2 \) is minimized, where \( w_s \) 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 \( L_{\text{tar}}(i) \) defined in Eq (16). Then, the data for each training speaker \( i \) can be repeated \( r_i = \text{round}(N_r \times w_i) \) 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 \( \epsilon \) will be equally weighted. However, because each sample is repeated \( r_i \) times and same error \( \epsilon_i(r) \) is obtained for each repetition \( r \), total error contributed by speaker \( i \), \( \epsilon_i \), is equal to \( r_i||\epsilon_i(r)||^2 \) 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.

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, \( S_{\text{nn}} \), to the target speaker \( s_{\text{tar}} \). Euclidean distance, \((\mu_{\text{nn}} - \mu_{\text{tar}})^T(\mu_{\text{nn}} - \mu_{\text{tar}})\), is used as the similarity measure, where \( \mu_{\text{nn}} \) is the supervector of state means in the input language model of nearest training speaker. Similarly, \( \mu_{\text{tar}} \) is the supervector of the target speaker.

Once \( S_{\text{nn}} \) is selected, the state-map \( M_{\text{nn}} \) 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 \( M_{\text{tar}} \) is available for the target speaker. Here, the nearest-neighbour based state-mapping algorithm described above is used to find \( M_{\text{tar}} \).

Once the adaptation data is mapped to the states of the output language, computation of \( w_{s, \text{out}} \) is exactly the same as the intra-lingual adaptation case. The adaptation data-dependent variables \( S^{(s)}_{\text{c}} \) and \( N^{(s)}_{\text{c}} \) in Eq (6) are computed by mapping data to output language states using \( M_{\text{tar}} \). Then, \( w_{s, \text{out}} \) 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
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 hyperparameter of the PLS and rPLS algorithms was tuned using cross-validation. The \( N_r \) parameter of the WRPLS algorithm was empirically set to 100.

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, \( \alpha \), 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 hyperparameter \( \alpha \) 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 \( \alpha \) used in the experiments are given in Table I.

When \( \alpha \) increases, the possibility of overfitting decreases. However, if \( \alpha \) 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
CSMAPLR and CMLLR performed equally well for the MGC both for MGC and LF0 and for all adaptation data sizes. The proposed NN-based state-mapping methods. The proposed NN-based state-mapping algorithm is compared with the other algorithms in Figure 8. CSMAPLR degraded the performance for the high-dimensional MGC features 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.

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, WR-PLS). 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
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).

![Percentage](image)

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

![Percentage](image)

(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.

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-
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**


