SPEAKER ADAPTATION WITH MINIMAL DATA IN STATISTICAL SPEECH SYNTHESIS SYSTEMS

A Thesis

by

Amir Mohammadi

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
September 2014

SPEAKER ADAPTATION WITH MINIMAL DATA IN STATISTICAL SPEECH SYNTHESIS SYSTEMS

Approved by:

Professor Cenk Demiroğlu, Advisor
Department of Electrical and Electronics Engineering
Özyeğin University

Professor Göksenin Yaralıoğlu
Department of Electrical and Electronics Engineering
Özyeğin University

Professor Hasan Sözer
Department of Computer Science
Özyeğin University

Date Approved: 20 Aug 2014

To my family and friends.

ABSTRACT

Statistical speech synthesis (SSS) systems have the ability to adapt to a target speaker with a couple of minutes of adaptation data. Developing adaptation algorithms to further reduce the number of adaptation utterances to a few seconds of data can have substantial effect on the deployment of the technology in real life applications such as consumer electronics devices. The traditional way to achieve such rapid adaptation is the eigenvoice technique which works well in speech recognition but known to generate perceptual artifacts in statistical speech synthesis. Here, we propose three methods to both alleviate the quality problems of the baseline eigenvoice adaptation algorithm while allowing speaker adaptation with minimal data. Our first method is based on using a Bayesian eigenvoice approach for constraining the adaptation algorithm to move in realistic directions in the speaker space to reduce artifacts. Our second method is based on finding pre-trained reference speakers that are close to the target speaker and utilizing only those reference speaker models in a second eigenvoice adaptation iteration. Both techniques performed significantly better than the baseline eigenvoice method in the objective tests. Similarly, they both improved the speech quality in subjective tests compared to the baseline eigenvoice method. In the third method, tandem use of the proposed eigenvoice method with a state-of-the-art linear regression based adaptation technique is found to improve adaptation of excitation features.

ÖZETÇE

İstatistiksel ses sentezi (İSS) sistemleri birkaç dakikalık uyarlama verisi kullanarak hedef konuşmacının sesine uyarlama yapabilme yeteneğine sahiptir. Uyarlama için gereken konuşma sürelerini daha da aşağıya, birkaç saniyeye, düşürmek için geliştirilen uyarlama algoritmaları, teknolojinin tüketici elektroniği gibi gerçek hayattaki uygulamalarda yaygınlaşmasında önemli etkiye sahip olabilir. Bu tarz hızlı uyarlamayı başarmanın geleneksel yöntemi özses tekniğidir ki konuşma tanımada iyi çalışmaktadır fakat istatistiksel ses sentezinde algısal artifeksler ürettiği bilinmektedir. Burada, hem temel özses uyarlama algoritmasının kalite problemini giderebilecek hem de asgari veri kullanarak konuşmacı uyarlamayı sağlayacak üç yöntem önerdik. Birinci yöntemimiz uyarlama algoritmasını, artifeksleri azaltmak için konuşmacı uzayında realistik doğrultularda hareket ettirmek amacıyla sınırlamak için önerdiğimiz Bayes özses yaklaşımının kullanımına dayanan yöntemdir. İkinci metodumuz ise hedef konuşmacıya yakın, önceden eğitilmiş referans konuşmacıları bulmaya ve o referans konuşmacı modellerini ikinci bir özses uyarlama iterasyonunda kullanmaya dayanır. Her iki teknik de nesnel testlerde temel özses metodundan önemli ölçüde daha iyi sonuçlar verdi. Benzer şekilde, her ikisi de temel özses metoduyla kıyaslandığında öznel testlerde ses kalitesini arttırdı. Üçüncü metodda, önerilen özses metodu ile son teknoloji doğrusal regresyon tekniğinin ardışık kullanımının uyarım özniteliklerinin uyarlanmasını geliştirdiği görüldü.

ACKNOWLEDGEMENTS

I would like to express my gratitude towards my supervisor, Professor Cenk Demiroğlu, for mentoring me through the learning process of this master thesis. His invaluable guidance and intimate attitude during this period has made me a much more solid researcher. I also would like to thank my thesis committee members Professor Göksenin Yaralıoğlu and Professor Hasan Sözer for spending their precious times. During the period of this thesis, many friends have been helpful to color my life. My past and present friends in the Speech Processing Lab. made this journey more fun. I have been blessed with a friendly and cheerful group of fellow students in the graduate school. I would also would like to thank my partner in life, Zohreh Mostaani, for her unconditional love and support. Special thanks to our brand new university for giving us an opportunity to work with wonderful people in a warm environment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Text-to-speech (TTS) is the process of synthesizing artificial speech for a given input text. It has been widely used in many application such as: e-book readers, navigation systems, voice-to-voice communication systems [1]. Typical TTS systems have two main components, text analysis and speech waveform generation, which are sometimes called frontend and backend, respectively. In the text analysis component, given input text is converted into a linguistic specification consisting of elements such as phonemes. In the speech waveform generation component, speech waveforms are generated from the produced linguistic specification [1]. There are two common approaches to TTS which are statistical speech synthesis (SSS) and concatenative speech synthesis (CSS).

In CSS, speech is segmented into smaller units and these units are kept in a database. During synthesis, the units that match the input text are selected and an utterance is synthesized by concatenating the selected units[2, 3]. Usually, A cost function and Viterbi search algorithm is used to select a unit such that there are no speech quality problems. There are two common problems with CSS: one is the defining a cost function so that the speech quality increases and the other one is unit definition and database size. In CSS, a speech segment is synthesized by playing back a waveform with matching text. An utterance is synthesized by concatenating several speech fragments [2].

Statistical speech synthesis (SSS) has proven to be a promising approach in text to speech (TTS) applications with some advantages compared to the concatenative approach [3]. An important advantage of the SSS approach is the ability to adapt to a target speaker with a couple of minutes of adaptation data [4]. Thousands of

voices have been generated with SSS using speech databases prepared for speaker-independent speech recognition systems in addition to freely available databases with different microphone types [5].

Although linear-regression based speaker adaptation in the SSS systems have been shown to be successful with only a couple of adaptation utterances [6], the issue of adaptation with a few seconds of data has not been investigated as much. High performance speaker adaptation with such minimal data can enable wider deployment of the technology especially in embedded devices where there may not be enough resources to store the utterances and/or users are not willing to train the system even with a couple of utterances.

Constrained maximum likelihood linear regression (CMLLR) [7, 8] or and constrained structural maximum a posteriori linear regression (CSMAPLR) methods can be used for rapid adaptation [6]. CSMAPLR method is more robust in small adaptation data sizes since it uses a prior distribution. Since there is no conjugate prior in the case of CMLLR, count smoothing technique has been proposed where an initial adaptation is first done using a rapid adaptation technique such as vocal tract length normalization (VTLN) [9]. Sufficient statistics for computing the transformation matrices are then smoothed by an interpolation of the statistics computed with the adapted model and the output of the rapid adaptation algorithm at each iteration.

Eigenvoice techniques have been traditionally used for rapid adaptation in speech recognition systems [10] but have also been investigated for SSS [11]. Eigenvoice adaptation can be implemented using different methods. One of the more successful methods is the Cluster Adaptive Training (CAT) [12] which has been used for SSS in [13, 14]. In [15], CAT has been used for creating the average voice when multiple corpora are used in training. Moreover, CAT was also used for rapid adaptation and shown to be more successful than CMLLR in [15]. Speaker adaptive training (SAT), which is a related technique, has also been used for SSS in [16, 17, 18].

Eigenvoices generated by those algorithms capture the most important and most common variations in speech. However, characteristics of the target speaker's voice that are not captured by the eigenvoices can also be important for speaker similarity. Moreover, perceptual artifacts are observed in synthesized speech after adaptation using eigenvoice approach with minimal data [11].

Interpolation between different speakers and styles have been used to generate voices with the desired voice quality, style, and emotion [19, 20]. In that approach, weights of pre-trained voices/styles are adjusted to make the interpolated voice sound close to target. This is similar to the eigenvoice approach except adaptation is done by interpolating the speaker-adapted voices, as opposed to eigenvoices, and weights are set manually.

We propose three methods to both alleviate the speech quality problems of the eigenvoice technique and improve the speaker similarity after adaptation with minimal data. The first method is based on using a Bayesian approach to estimate the weights of the eigenvoices with the goal of forcing the adaptation algorithm to move in realistic directions in the speaker space. Pre-training many speakers and using their models to empirically estimate the parameters of the prior distribution, which are then used in weight estimation, allows the system to create models with significantly less artifacts compared to the maximum likelihood (ML) based eigenvoice method.

Working in a target-independent eigenspace, novel speaker-specific directions that can be important for capturing the speaker characteristics may not be represented in the final model. The second proposed method is based on finding a set of k nearest-neighbors (NN) [21] from a set of pre-trained models after the first adaptation iteration. In a second iteration, only the closest k-NN's are used for finding the eigenspace. This approach allows using a speaker pool that is specific to the target speaker during eigenvoice training. To reduce the high computational complexity in training, a constrained training algorithm is proposed that uses sufficient statistics collected in the previous training iterations.

In eigenvoice adaptation techniques, new models are coarsely estimated with a few parameters. However, when the size of adaptation data increases, performance of eigenvoice adaptation saturates quickly. In the third proposed method, tandem eigenvoice/linear-regression approach is used to keep improving the adapted model with increased data sizes. To that end, we propose an additional step of linear regression adaptation after the eigenvoice adaptation step. Results showed significant improvements in the adaptation of log-fundamental frequency (LF0) features because those have low dimensionality which enables reliable training of linear regression matrices with only a few seconds of adaptation data.

Subjective experiment results show that the Bayesian eigenvoice method does not have the perceptual artifacts that are sometimes produced by the ML-based eigenvoice adaptation. For the mel-generalized cepstral (MGC) features, Bayesian eigenvoice with k-NN outperformed all other algorithms in the objective tests. Similarly, the tandem approach has the best objective adaptation performance among all methods for the LF0 parameter. Both Bayesian eigenvoice with k-NN and tandem methods outperformed the baseline linear regression [6] algorithm in the subjective speaker similarity and speech quality tests.

This paper is organized as follows. Baseline rapid speaker adaptations methods are described in Chapter 2. Proposed algorithms are described in Chapter 3. Experiment results are presented and discussed in Chapter 4. Finally, conclusion is done in Chapter 5.

CHAPTER II

PREVIOUS WORK

2.1 Rapid Statistical Speaker Adaptation Methods

Speaker adaptation with small amounts of data is typically done with linear regression based methods such as constrained maximum likelihood linear regression (CMLLR) and constrained structural maximum a posteriori linear regression (CSMAPLR). When the amount of adaptation data is minimal, eigenvoice-based methods can also be used since they have substantially lower number of parameters to learn.

CSMAPLR algorithm, which is state-of-the-art in the HMM-based TTS field [6], eigenvoice-based methods, and other methods are described below.

2.1.1 CMLLR and CSMAPLR

Most SSS systems model the speech unit using a N-state hidden semi-Markov model (HSMM). The emission pdf of the spectral parameters for each state c is modeled with a single Gaussian

$$p_c(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^L$ is the observation vector. Moreover, the duration of observation, d, is modeled with a Gaussian distribution.

$$b_c(d) = \mathcal{N}(d; m_c, \sigma_c^2) \tag{2}$$

where m_c and σ_c^2 are the mean and the variance, respectively.

After adaptation with linear regression, the new emission pdf is another Gaussian with

$$\hat{\boldsymbol{\mu}}_{c,lr} = \boldsymbol{A}\boldsymbol{\mu}_c + \boldsymbol{b} \tag{3}$$

and

$$\hat{\boldsymbol{\Sigma}}_{c,lr} = \boldsymbol{H}\boldsymbol{\Sigma}_c\boldsymbol{H}^T \tag{4}$$

where $\boldsymbol{A} \in \mathbb{R}^{L \times L}$ is the transformation matrix for the mean vector, $\boldsymbol{b} \in \mathbb{R}^{L}$ is the bias term, and $\boldsymbol{H} \in \mathbb{R}^{L \times L}$ is the transformation matrix for the covariance matrix. The duration models can also be adapted in the same manner.

To reduce the number of parameters, hence the amount of data required for reliable estimation of the parameters, the transformation matrices of the mean vector and the covariance matrix are tied together in the constrained linear transformation approach. The distribution is then

$$p_c(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{A}\boldsymbol{\mu}_c + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}_c\boldsymbol{A}^T). \tag{5}$$

In the CMLLR method, ML-based estimation is used for obtaining the transformation matrices [7, 8, 22]. While in the CSMAPLR method, a Bayesian approach is used to estimate the constrained linear regression parameters which is especially useful when there is limited amount of training data [6]. In this approach

$$\hat{\boldsymbol{\Lambda}} = \underset{\boldsymbol{\Lambda}}{\operatorname{argmax}}\, p(\boldsymbol{x}|\lambda, \boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \tag{6}$$

where $p(\boldsymbol{\Lambda})$ is the prior distribution of the transformation parameters $\boldsymbol{\Lambda}$, and λ is the parameter set of the Gaussians in the SSS model. Matrix variate normal distribution are used as the prior distribution $P(\boldsymbol{\Lambda})$:

$$P(\boldsymbol{\Lambda}) \propto |\boldsymbol{\Omega}|^{-\frac{L+1}{2}}|\boldsymbol{\Psi}|^{-\frac{L}{2}} \times exp\left[-\frac{1}{2}tr(\boldsymbol{W}-\boldsymbol{B})^T\boldsymbol{\Omega}^{-1}(\boldsymbol{W}-\boldsymbol{B})\boldsymbol{\Psi}^{-1}\right] \tag{7}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{L \times L}$, $\boldsymbol{\Psi} \in \mathbb{R}^{(L+1)\times(L+1)}$, and $\boldsymbol{B} \in \mathbb{R}^{L \times (L+1)}$ are the hyper-parameters of the prior distribution. Because prior is taken into account in estimation, parameter over-fitting because of data sparsity can be eliminated with the CSMAPLR algorithm. However, success of the algorithm depends on the proper selection of the prior distribution and its hyper-parameters.

In the CSMAPLR approach, priors are estimated using a hierarchical approach embedded into a tree structure [23]. First a global transformation matrix is estimated at the root node where its prior is an identity matrix. Then, the estimated transformation matrix is used as a prior for its child nodes, new transformation matrices are estimated and this process is further propagated down to their child

nodes. Imposing such a structure on prior estimation allows reliable estimation of the prior distribution which is especially important in limited adaptation data case since the posterior relies more on the prior pdf than the likelihood function in that case. In the CSMAPLR estimation, the hyper-parameter $\boldsymbol{\Psi}$ is fixed to the identity matrix and $\boldsymbol{\Omega}$ to a scaled identity matrix, $\boldsymbol{\Omega} = \tau \boldsymbol{I}_L$. τ is a positive scalar that controls the scale factor for the prior propagation and $\boldsymbol{I}_L$ is $L \times L$.

2.1.2 Eigenvoice Adaptation

Eigenvoice approach has been used for rapid adaptation in speech recognition and SSS [10, 11]. The idea is to find a set of R vectors in the high-dimensional space $\mathbb{R}^n$ ($n >> R$) that can be used to approximate a set of vectors in $\mathbb{R}^n$ by optimizing a distance measure. One way to accomplish this is using principal components analysis (PCA) that finds the directions in $\mathbb{R}^n$ where the data has the highest variance and the L_2 norm of the approximation error is minimum after projection. Solution with PCA are the eigenvectors of the sample covariance matrix with the highest eigenvalues.

In the context of SSS, each eigenvector is called an eigenvoice. The supervector for speaker s can be created by $\boldsymbol{\mu}^{(s)} = [\boldsymbol{\mu}_1^{(s)}\ \boldsymbol{\mu}_2^{(s)}\ ...\ \boldsymbol{\mu}_{N_{st}}^{(s)}]$ where N_{st} is the total number of states in all decision trees in the acoustic model.

In the eigenvoice approach, given a set of R eigenvectors $\boldsymbol{e}_r \in \mathbb{R}^n$, the original supervector for speaker s is represented as

$$\boldsymbol{\mu}^{(s)} = \boldsymbol{\mu}_{\text{SI}} + \boldsymbol{E}\boldsymbol{w}_s + \boldsymbol{\epsilon}_s \tag{8}$$

where $\boldsymbol{E} = [\boldsymbol{e}_1\ \boldsymbol{e}_2\ ...\ \boldsymbol{e}_R]$, $\boldsymbol{w}_s$ is weight vector of the speaker s, and $\boldsymbol{\epsilon}_s$ is the approximation error. Although $\boldsymbol{E}$ can be found by using the PCA method, it can also be estimated from the training data using a maximum-likelihood (ML) approach. One popular algorithm to do that is the Cluster Adaptive Training (CAT) technique which is really an adaptive training algorithm but can also be used for eigenvoice adaptation. In the case of CAT, columns of $\boldsymbol{E}$ can be seen to represent the clusters in the training data, and the weights for a given speaker are

the interpolation factors between those clusters.

An iterative algorithm is proposed in [12] for learning $\boldsymbol{E}$ from a training dataset. In the first step, $\boldsymbol{E}$ is initialized randomly. Then, weights are estimated using a maximum-likelihood approach for each speaker. Using those estimated weights, $\boldsymbol{E}$ is re-estimated and the whole procedure is repeated until convergence.

Although the algorithm in [12] is similar to the Expectation-Maximization (EM) algorithm, it is not EM because posterior distribution of the weights, hence the uncertainty in the weights, are not taken into account in the iterations. This can cause problems especially when there is insufficient data for some of the speakers. The algorithm proposed in [24] solves the problem by offering an exact EM solution. Here, the algorithm proposed in [12] is used for training $\boldsymbol{E}$ since there is sufficient data for each speaker during training.

In the ML-based CAT approach, given some adaptation data $\chi_a = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N_{o,s})}\}$, $N_{o,s}$ is the total number of observations from speaker s, the likelihood function

$$p(\chi_a|\boldsymbol{w}_s, \boldsymbol{E}) \propto$$
$$exp(-\frac{1}{2}\sum_{c=1}^{N_{st}}\sum_{i=1}^{N_c^{(s)}}(\boldsymbol{x}_c^{'(i)} - \boldsymbol{E}_c\boldsymbol{w}_s)^T\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}_c^{'(i)} - \boldsymbol{E}_c\boldsymbol{w}_s)) \quad (9)$$

where $\boldsymbol{E}_c$ is the c^{th} block of the $\boldsymbol{E}$ matrix corresponding to state c, $\boldsymbol{x}_c^{'(i)} = \boldsymbol{x}_c^{(i)} - \boldsymbol{\mu}_c$, $\boldsymbol{x}_c^{(i)}$ is 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 number of observations aligned with state c for speaker s.

After removing terms that are independent of $\boldsymbol{w}$ and $\boldsymbol{E}$, the objective function

$$O = (-\tfrac{1}{2}\textstyle\sum_{c=1}^{N_{st}} S_{xx,c}) + \boldsymbol{w}_s^T\boldsymbol{G}_w^{(s)}\boldsymbol{w}_s - \boldsymbol{w_s}^T\tfrac{1}{2}\boldsymbol{k}_w^{(s)} \quad (10)$$

where

$$S_{xx,c} = \sum_{i=1}^{N_c^{(s)}} \boldsymbol{x}_c^{'(i)^T}\boldsymbol{\Sigma}_c^{-1}\boldsymbol{x}_c^{'(i)}. \quad (11)$$

$$\boldsymbol{G}_w^{(s)} = \sum_{c=1}^{N_{st}} N_c^{(s)}\boldsymbol{E}_c^T\boldsymbol{\Sigma}_c^{-1}\boldsymbol{E}_c \quad (12)$$

$$\boldsymbol{k}_w^{(s)} = \sum_{c=1}^{N_{st}} \boldsymbol{E}_c^T \boldsymbol{\Sigma}_c^{-1} \boldsymbol{S}_{x,c}^{(s)} \tag{13}$$

$$\boldsymbol{S}_{x,c}^{(s)} = \sum_{i=1}^{N_c^{(s)}} \boldsymbol{x}_c^{'(i)} \tag{14}$$

Instead of jointly maximizing $\boldsymbol{E}$ and $\boldsymbol{w}_s$ or using the EM algorithm, an iterative algorithm is used to maximize the likelihood function [12]. In this approach, weight vector of speaker s, $\boldsymbol{w}_s \in \mathbb{R}^R$, is calculated as follows . Firstly, $\boldsymbol{E}$ is fixed to a constant matrix. Then, the objective function is maximized with respect to $\boldsymbol{w}_s$ and

$$\hat{\boldsymbol{w}}_{\text{CAT}} = {G_w^{(s)}}^{-1} k_w^{(s)} \tag{15}$$

Once the $\boldsymbol{w}_s$ vectors are computed for each speaker, they can be fixed, and the $\boldsymbol{E}$ matrix can be estimated. In this case, the objective function

$$\sum_{s=1}^{S}(-\frac{1}{2}\sum_{c=1}^{N_{st}} S_{xx,c}) + \boldsymbol{w}_s^T \boldsymbol{G}_w^{(s)} \boldsymbol{w}_s - {\boldsymbol{w}_s}^T \frac{1}{2} \boldsymbol{k}_w^{(s)} \tag{16}$$

is maximized with respect to $\boldsymbol{E}_c$, and

$$\hat{\boldsymbol{E}}_{c,\text{CAT}} = \boldsymbol{G}_c^{-1} \boldsymbol{K}_c \tag{17}$$

where

$$\boldsymbol{G}_c = \sum_{s=1}^{S} N_c^{(s)} \boldsymbol{w}_s \boldsymbol{w}_s^T \tag{18}$$

$$\boldsymbol{K}_c = \sum_{s=1}^{S} \sum_{i=1}^{N_c^{(s)}} \boldsymbol{w}_s \boldsymbol{x}_c^{'(i)^T} \tag{19}$$

and S is the total number of speakers. The new estimate of $\boldsymbol{E}$ can then be used to estimate $\boldsymbol{w}_s$ for all training speakers. Estimates of $\boldsymbol{E}$ and $\boldsymbol{w}_s$ can be improved with more iterations until convergence.

2.1.3 Vocal tract length normalization (VTLN)

Vocal tract shapes are different between speakers and they cause a mismatch between the speaker's utterance and the model in HMM-based automatic speech

recognizer [25]. A frequency warping factor, α, has been intorduced in order to warp the frequency axis of speech signal to normalize the speaker's utterance. The warping factor is obtained by searching over a grid of 13 factors spaced evenly between $0.88 \leq \alpha \leq 1.12$. This roughly reflects the 25 % variation in vocal tract shapes between speakers. In [25], two procedures are introduced to estimate the α. One procedure is the following three-step process:

1. A preliminary transcription of the utterance, W, is obtianed using the normalized model $\boldsymbol{\lambda}_N$ and the unwarped utterance $\boldsymbol{X}$.

2. $\hat{\alpha}$ is found as follows:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmax}} \, Pr(\boldsymbol{X}^{\alpha} | \boldsymbol{\lambda}_N, W) \tag{20}$$

   where $\boldsymbol{X}^{\alpha}$ is the transformed utterance by α.

3. Final recognition trancsription is obtianed using the utterance $\boldsymbol{X}^{\alpha}$ decoded with the model $\boldsymbol{\lambda}_N$.

2.1.4 Count Smoothing

Several rapid speaker adpatation methods use prior information to find robust transforms when minimal adpatation data is available. In [9], an initial adaptation is first done using a rapid adaptation technique such as VTLN [25, 26, 27] or PCMLLR [28, 29]. Then, output of the rapid adaptation algorithm is used to smooth the statistics for computing the transformation matrices in CMLLR.

2.1.5 CSMAPLR (VTLN)

In [30], VTLN is combined with the CSMAPLR algorithm. This approach improves the performance of the CSMAPLR algorithm in cases of small adaptation data in order of 1 utterance. As described in section 2.1.1, CSMAPLR uses a hierarchical approach embedded into a tree structure. Usually, the top global transformation matrix is calculated either using a maximum likelihood (ML) estimation or using a MAP approach with an identity matrix as a prior. In [30]

however, VTLN transformation is used as a prior for the top transformation estimation.

The VTLN transform can be viewed as

$$\boldsymbol{x}_\alpha = \boldsymbol{A}_\alpha \boldsymbol{x} \tag{21}$$

where $\boldsymbol{x}_\alpha = (\widetilde{x}_1, ..., \widetilde{x}_M)^T$ and $\boldsymbol{x} = (x_1, ..., x_L)^T$ are the warped and original observation vectors if we truncate them at M-th and L-th dimensions, respectively. $\boldsymbol{A}_\alpha$ is defined as [30]

$$A_{ml}(\alpha) = \frac{1}{(l-1)!} \sum_{n=max(0,l-m)}^{l} \binom{l}{n} \times \frac{(m+n-1)!}{(m+n-l)!} (-1)^n \alpha^{2n+m-l} \tag{22}$$

where $A_{ml}(\alpha)$ is the m-th row and the l-th column element of warping matrix $\boldsymbol{A}_\alpha$ and α is the warping factor. $\boldsymbol{A}_\alpha$ may also be directly applied to the dynamic features, where the transformation matrix is block diagonal with repeating $\boldsymbol{A}_\alpha$ matrix:

$$\boldsymbol{B}_\alpha = \begin{bmatrix} \boldsymbol{A}_\alpha & 0 & 0 & 0 \\ 0 & \boldsymbol{A}_\alpha & 0 & 0 \\ 0 & 0 & \boldsymbol{A}_\alpha & 0 \end{bmatrix}. \tag{23}$$

In the SMAP criterion in the CSMAPLR estimation, the top transformation matrix, $\boldsymbol{\Lambda}_1$, is calculated either using an ML estimation or a MAP estimation with identity matrix as the prior. Then, $\boldsymbol{\Lambda}_1$ is used as the $\boldsymbol{B}$ hyper-parameter in MAP estimation of $\boldsymbol{\Lambda}_2$ (refer to Eq (7)). In [30], $\boldsymbol{B}_\alpha$ is used as the $\boldsymbol{B}$ hyper-parameter in MAP estimation of the top transformation matrix, $\boldsymbol{\Lambda}_1$.

This approach has been tested in matched and unmatched conditions of speaker adaptation and also in both TTS and ASR setups. This method has also been compared to a cascade algorithm where first a VTLN adaptation is done and then its output is used as an SI model for the CSMAPLR algorithm. The cascade algorithm showed no significant difference compared to the original CSMAPLR algorithm. Results showed significant improvements compared to the CSMAPLR with no prior especially in the cases of less than 10 utterances. In the case of HMM-based TTS, the proposed method improved naturalness and intelligibility

of HMM-based synthetic speech compared to that using the CSMAPLR without the VTLN prior. In the case of HMM-based ASR, the proposed methods improved the performance. Results showed significant improvements especially in cases of mismatched conditions in terms of age, gender, and recording environments.

CHAPTER III

PROPOSED ALGORITHMS

Three algorithms are proposed to improve the CAT-based eigenvoice adaptation. An overview of the algorithms are shown in Fig. 1 and their desciptions are given below.

3.0.6 Bayesian CAT

In the proposed Bayesian CAT (BCAT) approach, E matrix is trained using the CAT procedure described above. However, in the BCAT approach, the weight vector for a target speaker s is estimated with the objective function

$$\hat{\boldsymbol{w}}_{\text{BCAT}} = \underset{\boldsymbol{w}}{\text{argmax}}\, p(\chi_a|\boldsymbol{w})p(\boldsymbol{w}) \tag{24}$$

where $p(\boldsymbol{w})$ is the prior distribution and set to $\mathcal{N}(0, \boldsymbol{\Sigma}_w)$ here. Thus,

$$p(\boldsymbol{w}) = \frac{1}{\sqrt{(2\pi)^R|\boldsymbol{\Sigma}_w|}} exp(-\frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Sigma}_w^{-1}\boldsymbol{w}). \tag{25}$$

Note that a point estimate of $\boldsymbol{w}$ is found and therefore the proposed estimator is not fully Bayesian. The term Bayesian is used here to indicate that the prior distribution is taken into account during estimation.

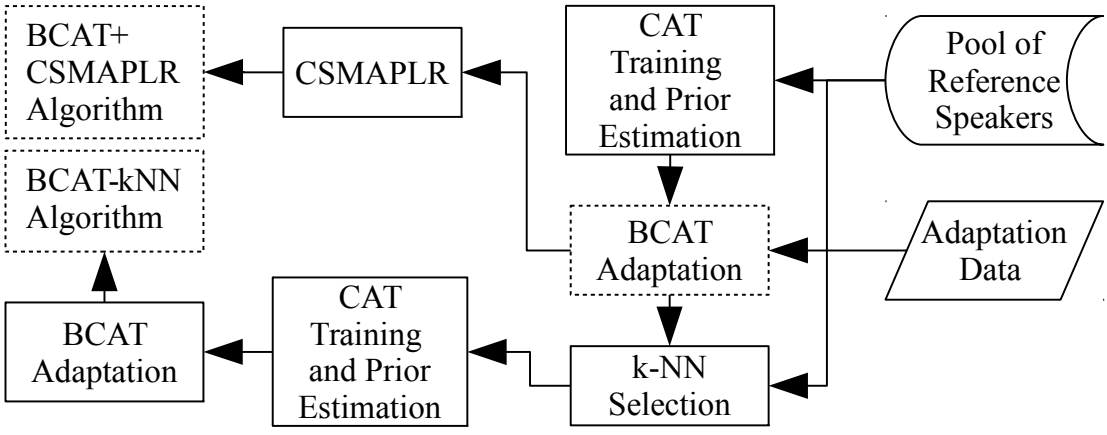

Figure 1: Overview of the proposed algorithms. Proposed algorightms are shown with dashed lines.

After removing the terms that are independent of $\boldsymbol{w}$ from the objective function, using Eq (9) to replace the likelihood term $p(\chi_a|\boldsymbol{w}_s)$ and with some matrix manipulation, the BCAT objective function becomes

$$\hat{\boldsymbol{w}}_{\text{BCAT}} = \underset{\boldsymbol{w}}{\text{argmax}}\, exp(\boldsymbol{w}^T\boldsymbol{E}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_x - \frac{1}{2}\boldsymbol{w}^T\boldsymbol{E}^T\boldsymbol{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{E}\boldsymbol{w})exp(-\frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Sigma}_w^{-1}\boldsymbol{w}). \quad (26)$$

where the block diagonal $\boldsymbol{\Sigma}^{-1} = diag(\boldsymbol{\Sigma}_1^{-1}, \boldsymbol{\Sigma}_2^{-1}, ..., \boldsymbol{\Sigma}_{N_{st}}^{-1})$, $\boldsymbol{S}_x = [\boldsymbol{S}_{x,1}, \boldsymbol{S}_{x,2}, ..., \boldsymbol{S}_{x,N_{st}}]$, and $\boldsymbol{N} = diag(N_1\boldsymbol{I}_{FxF}, N_2\boldsymbol{I}_{FxF}, ..., N_{N_{st}}\boldsymbol{I}_{FxF})$ where F is the size of the feature vectors.

The objective function can be maximized by noting that the posterior distribution $p(\boldsymbol{w}|\chi_a)$ is a Gaussian since the Gaussian distribution is the conjugate prior of the Gaussian likelihood function with unknown mean in Eq (9). Therefore, Eq (24) can be written as

$$\hat{\boldsymbol{w}}_{\text{BCAT}} = \underset{\boldsymbol{w}}{\text{argmax}}\, exp(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu}_{w|\chi})^T\boldsymbol{\Sigma}_{w|\chi}(\boldsymbol{w} - \boldsymbol{\mu}_{w|\chi})) \quad (27)$$

By completing the squares and using Eq (26),

$$\boldsymbol{\Sigma}_{w|\chi} = (\boldsymbol{E}^T\boldsymbol{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{E} + \boldsymbol{\Sigma}_w^{-1}), \quad (28)$$

and

$$\boldsymbol{\mu}_{w|\chi} = \boldsymbol{\Sigma}_{w|\chi}^{-1}\boldsymbol{E}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_x. \quad (29)$$

BCAT estimate of $\boldsymbol{w}$, $\hat{\boldsymbol{w}}_{\text{BCAT}}$, is the mean $\boldsymbol{\mu}_{w|\chi}$ of the posterior distribution. Hence,

$$\hat{\boldsymbol{w}}_{\text{BCAT}} = (\boldsymbol{E}^T\boldsymbol{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{E} + \boldsymbol{\Sigma}_w^{-1})^{-1}\boldsymbol{E}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_x. \quad (30)$$

$\boldsymbol{\Sigma}_w^{-1}$ is the hyperparameter of the prior distribution. It is used to enforce the adaptation algorithm to move more in specific directions compared to other directions. The idea is to learn the typical directions in the speaker space that the speaker-dependent models move during adaptation and use that as a prior information in adaptation when minimal observations are available.

$\boldsymbol{\Sigma}_w^{-1}$ is estimated from the data as follows. $\boldsymbol{w}$ is estimated for a set of speakers with large number of utterances per speaker using the ML approach. Because a

large number of utterances are used, there is no significant difference between ML or Bayesian estimation of weights with CAT. Then, $\boldsymbol{\Sigma}_w^{-1}$ is calculated by using the sample covariance matrix of the weights and setting the off-diagonal elements to 0.

In the SSS approach, different decision trees are used for the LF0 and MGC features [31]. Thus, the eigenspace of those two features are modelled independently in this work. In CAT, different decision trees can be used for different clusters [32]. Here, we followed the implementation in [12] where same decision-tree structures are used for all clusters.

Problem of overfitting with the ML-based CAT approach when the data is scarce has also been observed in the context of expressive speech synthesis [32]. To address the problem, a count smoothing approach is proposed where the statistics $\boldsymbol{G}_w$ and $\boldsymbol{k}_w$ in Equations (12) and (13) are smoothed using

$$\hat{\boldsymbol{G}}_w = \boldsymbol{G}_w + \tau \frac{\boldsymbol{G}_w^{(pri)}}{\sum_{c=1}^{N_{st}} N_c^{(pri)}} \tag{31}$$

$$\hat{\boldsymbol{k}}_w = \boldsymbol{k}_w + \tau \frac{\boldsymbol{k}_w^{(pri)}}{\sum_{c=1}^{N_{st}} N_c^{(pri)}} \tag{32}$$

where $\boldsymbol{G}_w^{(pri)}$ and $\boldsymbol{k}_w^{(pri)}$ are prior statistics, $N_c^{(pri)}$ is the number of frames aligned with component c in the training data that is used for computing the prior statistics, and τ is used for tuning the weight of the priors. A set of discrete labels for expressiveness were derived in [32] and $\boldsymbol{G}_w^{(pri)}$ is computed using training data that has the same label as the adaptation data.

Count smoothing technique has also been used for speaker adaptation in [9] where the CMLLR method is used and the statistics that are required to compute the linear transformations in CMLLR are smoothed. Smoothing is done using an interpolation of the prior statistics computed with a rapid adaptation algorithm such as vocal tract length normalization (VTLN) and adaptive statistics computed with the CMLLR method. In the BCAT approach, a prior distribution for the weight vector $\boldsymbol{w}$ is used instead of smoothing the statistics.

Using a prior distribution for the weight vector in eigenvoice adaptation has

been proposed in [33] in the context of voice conversion using Gaussian mixture models (GMM). In that case

$$\hat{\boldsymbol{w}} = \left\{\tau\boldsymbol{\Sigma}_w + \boldsymbol{G}_w\right\}^{-1}\left\{\tau\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\mu}_w + \boldsymbol{k}_w\right\} \tag{33}$$

where τ is used for tuning and the prior pdf has a mean of $\boldsymbol{\mu}_w$ and covariance of $\boldsymbol{\Sigma}_w$. The form of the solution is similar to count smoothing where the hyperparameters $\boldsymbol{\Sigma_w}$ and $\boldsymbol{\mu_w}$ are used for smoothing the statistics. In BCAT, $\boldsymbol{\mu}_w$ is set to $\mathbf{0}$ so that the adapted model does not significantly deviate from the average voice model if there is not enough adaptation data. In that case, $\boldsymbol{k}_w$ is not smoothed and, assuming τ is equal to 1, Equation 33 becomes equivalent to Equation 30. However, in the GMM case, probabilistic alignment of Gaussians with speech frames is done which leads to a solution with the iterative EM algorithm [33]. In SSS, each state is represented with a single Gaussian and, here, state-level forced alignment of the adaptation audio with the corresponding text is used to map the Gaussians to speech frames. Moreover, observing the fact that the Gaussian distribution is the conjugate prior of the likelihood function, a closed-form maximum a posteriori (MAP) solution is derived. Furthermore, we also propose estimating target-specific $\boldsymbol{\Sigma}_w$ parameter in the next section.

In [24], another algorithm is proposed for eigenvoice MAP adaptation of GMM to a target speaker using a Gaussian prior. In that approach, a zero-mean Gaussian distribution with identity covariance matrix is used both during training and adaptation. In that case

$$\hat{\boldsymbol{w}} = \left\{\boldsymbol{I} + \boldsymbol{G}_w\right\}^{-1}\left\{\boldsymbol{k}_w\right\}. \tag{34}$$

The approach in [24] is developed assuming that there is limited amount of training data for each speaker during training. In those cases, to avoid overfitting, it is important to use a prior distribution which regularizes the estimation of $\boldsymbol{E}$. However, there is enough training data for each speaker in our case and ML estimation of $\boldsymbol{E}$ is sufficient. Therefore, instead of imposing a fixed prior distribution

during training, ML training is used here and the hyperparameters of the prior distribution are emprically estimated after training is completed.

3.0.7 BCAT k-Nearest-Neighbor (BCAT k-NN) Approach

Iterative BCAT adaptation can be performed that uses the information learned in the previous iterations to further exploit the available information in the reference speakers. To that end, a k-NN approach is proposed here where $\boldsymbol{\Sigma}_w^{-1}$ and $\boldsymbol{E}$ matrices are trained in a target-specific way for better speaker adaptation.

Target specific parameters are trained as follows. After the first BCAT adaptation step, resulting model is used to find k nearest-neighbors (k-NN) from a large pool of reference speakers. Those neighbors are then used to create the $\boldsymbol{E}$ matrix and the covariance of the prior distribution $\boldsymbol{\Sigma}_w^{-1}$ as described in Section 3.0.6. Distance measures for finding the k-NN's are described in the next section.

Training the $\boldsymbol{E}$ matrix for each target speaker can be time-consuming. To solve the issue, we propose a constrained training algorithm where the covariance matrices of Gaussians $\boldsymbol{\Sigma}_c$ are not updated. Alignment of Gaussians are also kept constant throughout the iterations. To find $\boldsymbol{E}$ for a subset of speakers, $\boldsymbol{G}_c$ in Eq (18) and $\boldsymbol{K}_c$ in Eq (19) are needed. If the sufficient statistics

$$\boldsymbol{G}_{c,s} = N_c^{(s)} \boldsymbol{w}_s \boldsymbol{w}_s^T \tag{35}$$

$$\boldsymbol{K}_{c,s} = \sum_{i=1}^{N_c^{(s)}} \boldsymbol{w}_s \boldsymbol{x}_c^{\prime(i)^T} \tag{36}$$

for each speaker s are precomputed, then, $\boldsymbol{G}_{c,nn} = \sum_{s=1}^{S_{nn}} \boldsymbol{G}_{c,s}$ and $\boldsymbol{K}_{c,nn} = \sum_{s=1}^{S_{nn}} \boldsymbol{K}_{c,s}$ and the new $\boldsymbol{E} = \boldsymbol{G}_{c,nn}^{-1} \boldsymbol{K}_{c,nn}$.

Using the new $\boldsymbol{E}$ matrix, $\boldsymbol{w}_s$ can be computed for the nearest neighbors. The statistics $\boldsymbol{S}_{x,c}^{(s)}$ in Eq (13) does not depend on $\boldsymbol{E}$ and can be precomputed for each speaker. Rest of the calculations to find $\boldsymbol{w}_s$ does not depend on the data and can be computed fast.

The algorithm above allows only single update to $\boldsymbol{E}$ and $\boldsymbol{w}$ and is not iterative. However, in our experiments, one iteration is enough for computing $\boldsymbol{E}$ and weight vectors in the BCAT-kNN case.

3.0.8 Nearest-Neighbor (NN) Selection

Success of BCAT k-NN technique depends on using a distance measure that should both correlate well with perception and be reliable when minimal speech is available. We explored several different distance measures that operates either by directly using the waveform or by first adapting with the CSMAPLR or the eigenspace method and then measuring the distance between the speaker-adapted model and the reference models.

N_r reference speakers are trained offline. The reference speakers that are closest to the target speaker are used in the BCAT-kNN approach. Two different set of reference speakers are selected for MGC and LF0 features. The distance measures that are investigated here are

3.0.8.1 L_1 and L_2 Norms

L_p norm of two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ is defined to be

$$L_p = (\sum_{i=1}^{n} |v_{1,i} - v_{2,i}|^p)^{1/p} \tag{37}$$

where $|.|$ is the absolute value operator. The problem with the L_p norm in our case is representing each speaker with a single vector. Here, we ignore the covariances of Gaussian distributions on each HMM state and instead concatenate the mean vectors to create a supervector for each speaker.

Different decision tree is used for each HMM state. A separate supervector is created for each tree. For tree t and speaker s, the supervector $\boldsymbol{s}^{(t)} = [\boldsymbol{\mu}_{1,s}^{(t)}; \boldsymbol{\mu}_{2,s}^{(t)}; ...; \boldsymbol{\mu}_{N_t,s}^{(t)}]$ where N_t is the number of leaf nodes in tree t. For L_p norm, the distance between t^{th} decision trees of speaker s and reference speaker s_r is then defined as

$$d_{s_r,s}^{(t)} = (\sum_{j=1}^{D_t} |\boldsymbol{s}_j^{(t)} - \boldsymbol{s}_{\boldsymbol{r}_j}^{(t)}|^p)^{1/p} \tag{38}$$

where D_t is the length of the supervector for tree t. Finally, the distance of a reference speaker to a target speaker s is found by

$$\frac{1}{N_{tr}} \sum_{t=1}^{N_{tr}} d_{s_r,s}^{(t)} \tag{39}$$

where N_{tr} is the total number of trees.

In CAT and BCAT methods, speakers can also be represented with low-dimensional $\boldsymbol{w}$ vectors. Those $\boldsymbol{w}$ vectors, instead of the supervectors, can also be used for computing the L_p distances. In fact, higher performance has been achieved when $\boldsymbol{w}$ vectors were used in distance computations.

3.0.8.2 Cosine Distance

Cosine distance of two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are defined to be

$$cos(\theta) = \frac{< \boldsymbol{v}_1, \boldsymbol{v}_2 >}{\|\boldsymbol{v}_1\| \|\boldsymbol{v}_2\|} \tag{40}$$

where $< \boldsymbol{v}_1, \boldsymbol{v}_2 >$ is the inner product of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Cosine distance is really the normalized correlation of two vectors. $\boldsymbol{w}$ vectors were used to compute the cosine distances.

3.0.8.3 RMSE distance

Instead of focusing on the emission pdf parameters for defining the distance, one can also resynthesize the adaptation utterances using the reference models similar to [34]. In that case, a distance measure is needed to compare the synthesized audio with the original audio. Here, the commonly used root mean square error (RMSE) is used for calculating the distances for MGC and LF0 features.

In the case of RMSE distance, the length of synthesized speech should be equal to the length of adaptation utterance. Moreover, state durations should also match to make a meaningful comparison. To resolve the issue, the adaptation utterance is first state-aligned with the average-voice model and the duration of each state is found. Then, during synthesis, MGC and LF0 parameters are generated with the reference models using the durations from the state alignment phase.

3.0.9 Tandem BCAT/CSMAPLR Approach

Adaptation flexibility of CSMAPLR algorithm is low when data is scarce because of the large number of free parameters in transformation matrices. When only seconds of data is available, many states are clustered together and only a few transformation matrices can be trained as shown in Table 1. This problem is further exacerbated with the prior distribution used in CSMAPLR which limits the distance that the adapted model with the CSMAPLR algorithm can move away from the Speaker Independent (SI) model. BCAT algorithm, however, has substantially lower number of parameters and can adapt more effectively in limited data case. Still, BCAT can only make coarse adaptation in predetermined directions and cannot refine the models when more data is available.

In the tandem approach, the BCAT algorithm is used first to rapidly approach to the target speaker. In the second step, output of the BCAT algorithm is used as the new SI model for the CSMAPLR algorithm as shown in Fig. 1. Because the new SI model is already close to the target, constraints imposed by the CSMAPLR prior becomes less important. Moreover, the CSMAPLR algorithm can move the model in directions that may not be possible with the BCAT model and refine the models of some of the states and get them closer to the target models when more data is available.

CHAPTER IV

EXPERIMENTS

We evaluated the speaker similarity performance of the proposed systems both with objective and subjective measures and compared with the baseline CAT and CSMAPLR algorithms. Moreover, speech quality of the proposed systems are also compared with the baseline systems using listening tests. The BCAT-kNN algorithm relies on selecting good NN's. Thus, we have investigated the performance of different distance measures described in Section 3.0.8 for selecting k-NN.

4.0.10 Experiment Setup

All systems in the experiments were trained with 78 dimensional vectors consisting of 24 Mel-Generalized Cepstrum Coeffients (MGCs), 1 log-energy, 1 log-F0 (LF0) coefficient, and their delta and delta-delta parameters. 20 msec analysis window with 5 msec frame rate is used for feature extraction. Phonemes are modelled with 5 state Hidden Semi-Markov Models (HSMM).

Wall Street Journal (WSJ1) database is used to train the average voice and the speaker-adapted voices. Four male speakers with 1250 utterances for each of them are used for training the average voice. For the proposed system, 136 male reference speakers from the WSJ database are trained using 150 utterances per speaker with CSMAPLR adaptation and an additional MAP adaptation. HTS 2.2 training and synthesis tools are used to generate the samples for the baseline systems [35]. Speaker adaptive training (SAT) is used during training the SI model and the reference models.

Root-mean-square-error (RMSE) is used for objectively measuring the distance between the MGC and LF0 features of synthesized and original speech samples. To make meaningful comparison between them, original speech is first aligned at

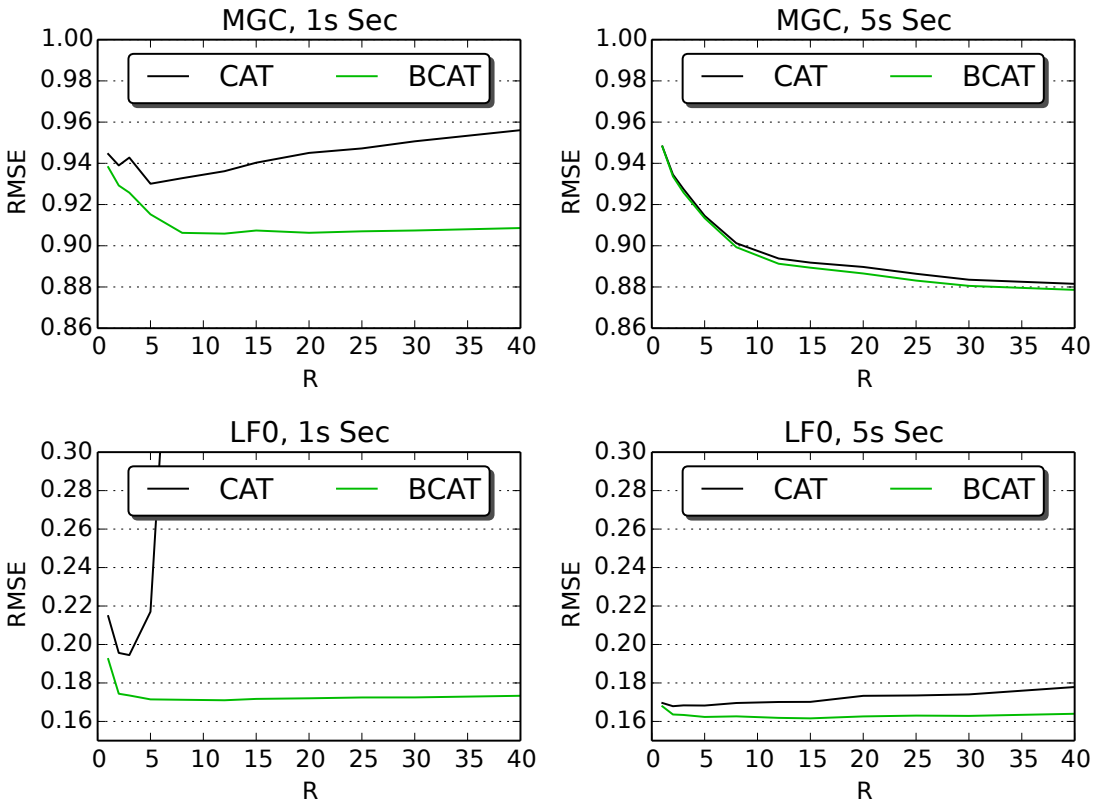

Figure 2: Performance of the CAT and BCAT algorithms for different ranks of the $\boldsymbol{E}$ matrix. Results are shown for the LF0 and MGC features using 1 second and 5 seconds of adaptation data.

the state level with the average voice model. Durations obtained during alignment is used for synthesizing the samples. For each target speaker, adaptation is performed for 1, 2, 3, 4, and 5 seconds of adaptation data, excluding any silence segments. 40 utterances are synthesized for each target speaker to measure the RMSE distance. 21 target speakers are used. For each target, a speaker-dependent (SD) model is generated using CSMAPLR adaptation with an additional MAP step using 150 adaptation utterances per speaker [6]. Those SD[1] models are used as the upper bound in adaptation performance.

Performance of CAT and BCAT with different rank values and adaptation sizes are measured to find the best possible ranks for CAT and BCAT at each data size. RMSE performances are shown for MGC and LF0 at 1 sec and 5 sec in

[1]In this work by *SD* we mean the speaker-adapted model obtained with large amount of adaptation data.

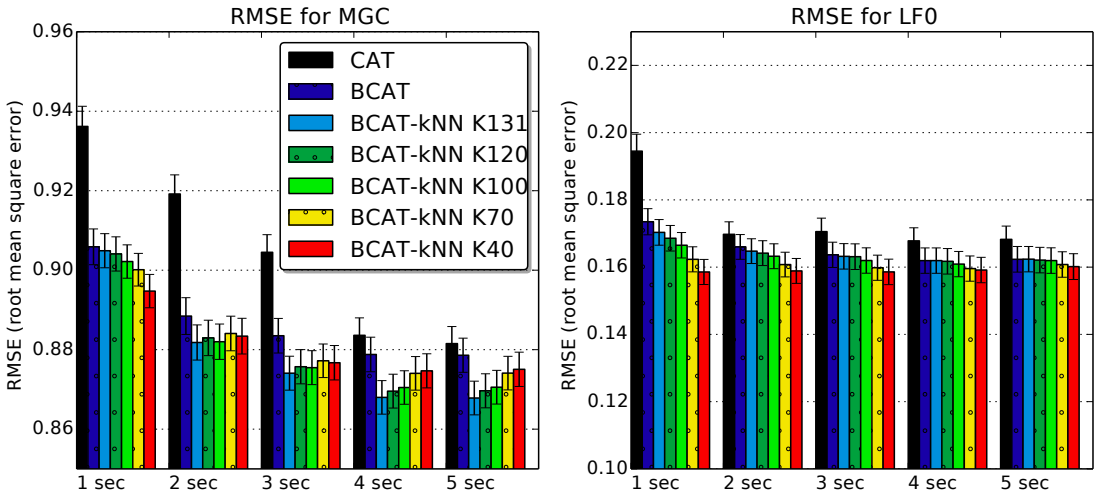

Figure 3: Performance of the BCAT-kNN algorithm for different k values. RMSE results are shown both for LF0 and MGC features with 95% confidence intervals.

Fig. 2. CAT has severe overfitting problem for the 1 sec case. However, for 5 sec, the difference between CAT and BCAT is less significant especially for the MGC feature. Based on the experiment results, rank of the $\boldsymbol{E}$ matrix was set to 8, 30, 30, 40, 40 for 1 through 5 seconds in MGC and 3, 3, 5, 5, 5 in LF0 adaptations. Same values are used for CAT and BCAT.

Another parameter to set is the k value in the BCAT-kNN algorithm. Similar to the rank of $\boldsymbol{E}$, the best value for k was found experimentally. Performance of BCAT k-NN for different k values are shown for MGC and LF0 features in Fig. 3. Based on those results k is set to 131 for MGC and 40 for LF0. For the MGC features, removing only 5 speakers from the full set of reference speakers was enough in all cases except the 1 second case. We believe that the five speakers are target-specific outliers and they cause misestimations in the ML-based CAT algorithm. Thus, removing them improves the performance. However, any further removal of reference speakers from the training set degrades the performance for MGC.

For the 1 second case, the effect of prior distribution becomes very important. Using 40 NN's for MGC instead of 131 reduces the variance of the prior which provides higher performance for that case. Moreover, difficulties in selecting the nearest-neighbors with only one second of adaptation data also contributes to the

problem for the MGC case as shown in Table 2. For LF0, a relatively small number of speakers were enough to train the eigenspace. Including more speakers into the training process unnecessarily increases the variability of the weight vectors which degrade the performance.

ABX test is used to subjectively measure the similarity of synthesized samples compared with the speaker's original samples. In the ABX test, listeners prefer sample A or sample B depending on perceived similarity to the reference sample X. A and B samples are synthesized from different adaptation methods randomly and X samples are "synthetic-copy" of the original recordings. Thus, the goal in the ABX test is to compare two systems and assess which one produces speech that is more similar to the target speaker. AB test is done to measure the quality differences. As opposed to the ABX test, a reference X sample is not needed in the AB test. Listeners prefer A or B sample depending on the perceived speech quality. Because a reference X sample is needed to test similarity to target speaker, different tests are used for quality and similarity assessment.

In listening tests, similar to the RMSE tests, for each target speaker, adaptation is performed with 1, 2, 3, 4, and 5 seconds of data. For each adaptation data size, one utterance is synthesized per target speaker. 10 target speakers are used. Eight listeners took the tests.

In CSMAPLR, the decision trees used in the average voice model training is also used for estimating multiple transformation matrices. Block diagonal transformation matrices are used. States occupancy thresholds are tuned to determine the optimal number transformation matrices. Results are shown in Table 1. Very few number of transformation matrices were found especially for MGC. For the 1 second case, sometimes there was no adaptation at all. For the rest of the cases, number of matrices increased only slightly. Note that, the adaptation algorithm becomes a MAP estimation of the global CMLLR transform when only one global transformation matrix is generated. Higher number of transformation matrices were available for the LF0 feature since the dimensionality of LF0 features are

substantially lower than MGC.

Table 1: RMSE and number of estimated transforms in CSMAPLR Adaptation with different threshold values. Minimum values in RMSE and selected thresholds are in bold.

		RMSE				
	Thresholds	1 sec	2 sec	3 sec	4 sec	5 sec
MGC	1000	**1.01**	0.99	0.98	0.97	**0.91**
	500	1.02	**0.98**	**0.94**	**0.92**	**0.91**
	250	1.03	0.99	0.97	0.93	0.92
	125	1.05	1.01	0.99	0.95	0.94
	65	1.07	1.03	1.02	0.97	0.96
LF0	200	**0.20**	**0.17**	0.17	**0.16**	**0.16**
	100	**0.20**	**0.17**	**0.16**	**0.16**	**0.16**
	50	**0.20**	**0.17**	**0.16**	**0.16**	**0.16**
	25	**0.20**	**0.17**	**0.16**	**0.16**	**0.16**
	13	0.21	**0.17**	**0.16**	**0.16**	**0.16**

		Average number of transforms				
	Thresholds	1 sec	2 sec	3 sec	4 sec	5 sec
MGC	1000	0.0	0.1	0.2	0.5	1.0
	500	0.3	0.9	1.0	1.4	2.9
	250	3.3	3.4	4.7	5.4	7.8
	125	9.2	12.5	16.1	22.0	25.3
	65	17.2	24.1	30.3	38.3	45.6
LF0	200	0.0	1.8	2.7	2.7	3.6
	100	2.2	3.6	5.7	11.5	17.5
	50	3.1	14.0	31.6	50.5	65.9
	25	23.4	56.2	97.5	136.0	170.5
	13	100.4	170.3	249.0	337.8	418.0

4.0.11 Distance Measures for NN Selection

L1 norm, L2 norm, cosine, and RMSE distances are compared for selecting a nearest-neighbor given a target speaker model. For L1 norm, L2 norm, and cosine distances, delta, delta-delta, and energy features were not used in the distance computations because those feature were found to degrade the performance. For CAT and BCAT methods, L1, L2, and cosine distance computations are done using both the supervectors and the $\boldsymbol{w}$ vectors.

Different combinations of distance measures and adaptation algorithms are compared and RMSE of selected nearest-neighbors to original samples of target

speakers for MGC and LF0 features are shown in Table 2. To pick the best possible nearest-neighbor, best NN, 20 utterances for each target speaker are synthesized with all the reference speakers and the reference speaker with the smallest RMSE is found. This procedure is done to find the upper limit in performance and the results are shown in Table 2 for comparison purposes.

$\boldsymbol{w}$ vectors computed with CAT and BCAT algorithms in conjunction with L1 and L2 distances are found to be the best measures to pick the nearest neighbors for MGC. In LF0, however, most L1 and L2 distances worked equally well. RMSE distance also performed as well as other methods to find the NN speakers for LF0 features. Cosine distance was not a good measure to find the NN speakers and it performed worse than others in all cases as shown in Table 2. Based on these results, BCAT-w L2, which is the L2 distance between BCAT-based $\boldsymbol{w}$ vectors of target and reference models, were used in the rest of the experiments since its performance is always better than the others both for MGC and LF0 features.

A 2-dimensional visualization of a target speaker and reference speakers is shown in Fig. 4. Low dimensional $\boldsymbol{w}$ vectors were used to represent each speaker which are projected to two dimensions using multidimensional scaling based on L2 distance. For a given target speaker, 50 nearest-neighbors are selected using 5 seconds of adaptation data and BCAT-w L2 method. Most of the closest reference speakers to target are successfully selected with only 5 seconds of adaptation data as shown in Fig. 4.

4.0.12 Objective Measure Tests

All adaptation algorithms are compared objectively and results are shown in Fig. 5 and Fig. 6. For comparison purposes, RMSE of the Speaker Independent (SI) model and Speaker Dependent (SD) models are also presented. SD models were created using CSMAPLR adaptation and an additional step of MAP adaptation with 150 utterances. Also, even though we are focused on cases where at most five seconds of adaptation data is available, performance with 10sec and 20sec of data are also shown. For the one minute case, only the performance of the CSMAPLR,

Table 2: RMSE comparison for MGC and LF0 when NN speakers are chosen by different distance measures. "-w" means that instead of mean supervectors, weight vectors of CAT-based adaptations were used. Best possible RMSE that could be achieved is shown in Best NN rows. Minimum values are in bold.

MGC	1 sec	2 sec	3 sec	4 sec	5 sec
Best NN	0.92	0.92	0.92	0.92	0.92
BCAT-w L2	**0.97**	**0.95**	**0.95**	**0.95**	**0.95**
BCAT-w L1	**0.97**	**0.95**	**0.95**	0.96	**0.95**
CAT-w L2	**0.97**	**0.95**	**0.95**	0.96	0.96
CAT-w L1	0.99	0.96	**0.95**	0.96	0.96
BCAT L1	1.01	0.97	0.97	0.98	0.97
RMSE distance	0.99	0.98	0.99	0.99	0.97
BCAT L2	1.01	0.98	0.98	0.99	0.99
CAT-w cosine	1.02	0.99	1.00	1.00	1.02
BCAT-w cosine	0.98	1.00	1.00	1.00	1.01
CAT L2	1.01	1.00	1.01	0.98	0.99
CAT L1	1.01	1.00	1.00	0.99	0.98
CSMAPLR L1	1.11	1.04	0.98	0.99	0.97
CSMAPLR L2	1.11	1.05	1.01	0.97	0.98
BCAT cosine	1.06	1.05	1.05	1.03	1.02
CSMAPLR cosine	1.09	1.05	1.04	1.04	1.04
CAT cosine	1.05	1.06	1.02	1.02	1.02
LF0	1 sec	2 sec	3 sec	4 sec	5 sec
Best NN	0.17	0.17	0.17	0.17	0.17
BCAT-w L2	**0.20**	**0.19**	**0.18**	**0.18**	**0.18**
CSMAPLR L1	**0.20**	**0.19**	**0.18**	**0.18**	**0.18**
RMSE distance	0.21	**0.19**	**0.18**	**0.18**	**0.18**
BCAT-w L1	**0.20**	**0.19**	0.19	**0.18**	0.19
BCAT L1	**0.20**	**0.19**	0.19	0.19	**0.18**
CAT L1	0.21	**0.19**	0.19	**0.18**	**0.18**
CAT L2	0.21	**0.19**	0.19	**0.18**	0.19
CSMAPLR L2	**0.20**	**0.19**	0.19	0.19	0.19
BCAT L2	**0.20**	0.20	0.19	0.19	**0.18**
CAT-w L1	0.22	0.20	0.19	0.19	0.19
CAT-w L2	0.21	0.20	0.19	0.19	0.19
CAT-w cosine	0.23	0.21	0.20	0.21	0.20
CAT cosine	0.23	0.23	0.21	0.21	0.20
BCAT-w cosine	0.29	0.26	0.23	0.21	0.23
CSMAPLR cosine	0.29	0.28	0.27	0.28	0.29
BCAT cosine	0.34	0.29	0.26	0.25	0.23

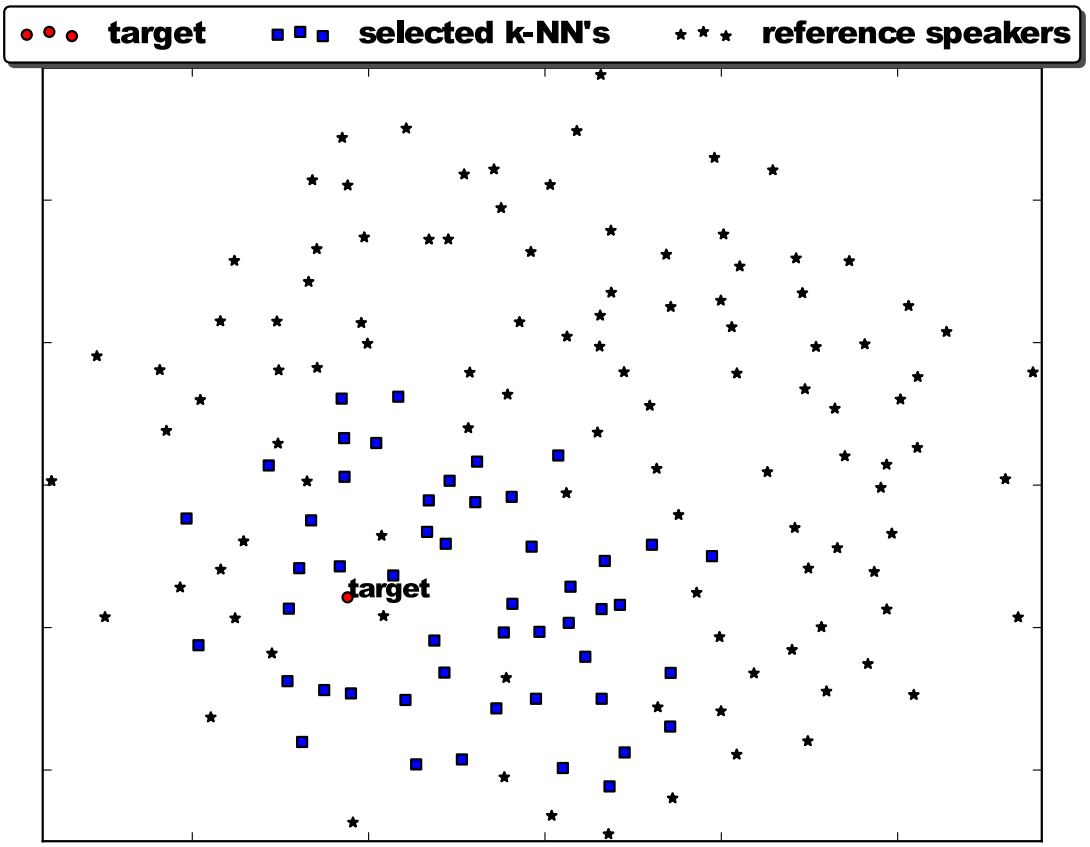

Figure 4: 2-D visualization of speakers based on their L2 distance of $\boldsymbol{w}$ vectors for the MGC features. 40 utterances were used to extract the $\boldsymbol{w}$ vectors with the BCAT approach. Also, 50 nearest-neighbors selected using 5 seconds of adaptation data from a target speaker are shown. L2 distance of $\boldsymbol{w}$ vectors were used to select the k-NN.

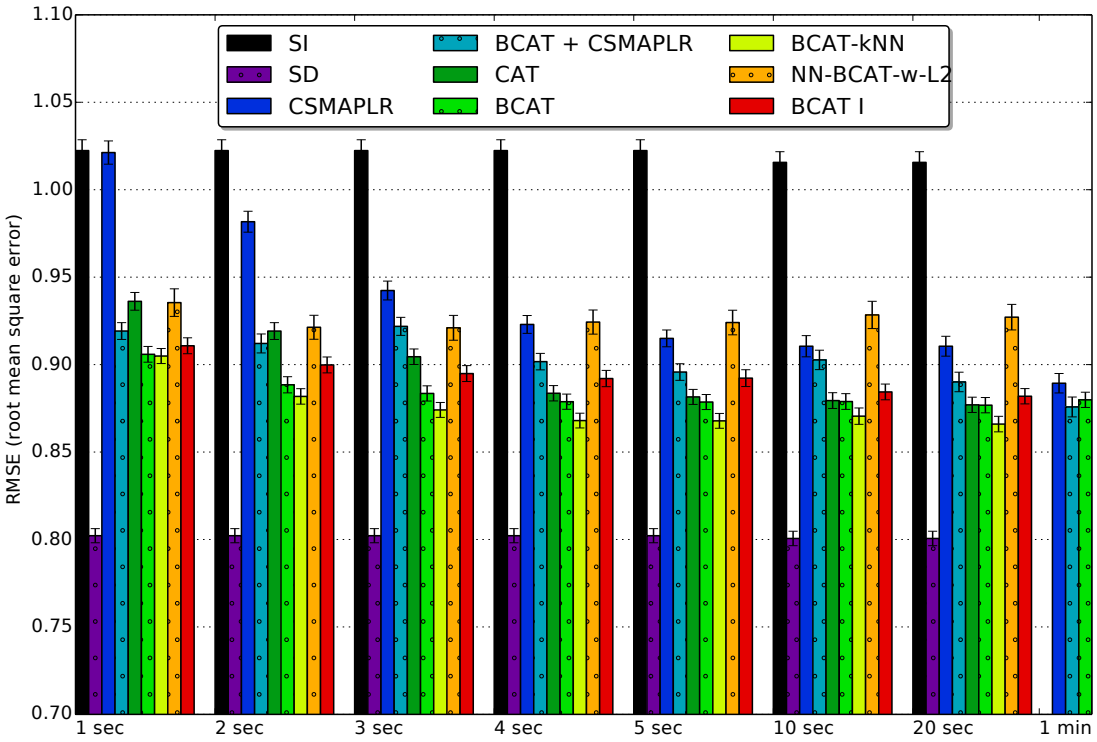

Figure 5: Objective evaluation (RMSE) of adaptation techniques for MGC features with 95% confidence intervals.

BCAT, and BCAT+CSMAPLR algorithms are shown to analyze the performance of the BCAT+CSMAPLR algorithm when more data becomes available. Performance of the other algorithms saturate after 5 seconds.

For MGC features, BCAT is better than CAT for all adaptation data sizes but as the amount of data increases their difference decreases as shown in Fig. 5. This is expected because with more data being available, the importance of using a prior distribution decreases. The difference between CAT and BCAT becomes insignificant when more than four seconds of adaptation data is used.

BCAT has also been compared with BCAT-I where the covariance of the prior is set to identity matrix. Using an identity matrix for prior degraded the performance of BCAT significantly except for 10 sec and 20 sec cases.

State-of-the-art CSMAPLR method did not perform as well as the eigenspace techniques since the data size is very limited. The number of transformation matrices generated with CSMAPLR is very low as shown in Table 1. Attempting to generate more transformation matrices creates rank deficiency problem because

of data scarcity and degrades the performance. Hence, the CSMAPLR cannot take advantage of tree-structured prior estimation for generating multiple transforms and has a lower performance compared to eigenspace methods. However, CSMAPLR performance continues to improve with more data as shown in Fig. 5.

One can also synthesize with the nearest-neighbor model without any further adaptation. The result with that approach is denoted with NN-BCAT-w-L2 which indicates that NN is selected using the $\boldsymbol{w}$ vector estimated with the BCAT algorithm using the L2 distance. Even though this approach is inferior to BCAT algorithm, it is better than the CSMAPLR algorithm when there is 1, 2, and 3 seconds of adaptation data.

For MGC features, performance of tandem BCAT/CSMAPLR (BCAT+CSMAPLR) was always better than CSMAPLR but worse than BCAT except for the 1 minute case. After evaluating the number of transformation matrices in both CSMAPLR and tandem BCAT/CSMAPLR, it was observed that both algorithms had the same number of transformation matrices. However, when the seed model to the CSMAPLR algorithm was the SI model, CSMAPLR lowered the RMSE. But when the seed model was the output of BCAT algorithm, the CSMAPLR algorithm increased the RMSE compared to its seed model. This shows that transforms estimated in CSMAPLR change our model to a poor estimate of the target model while still being a better model than the SI model. This is thought to be because of the rank deficiency problem in the CSMAPLR approach when only few seconds of data is available. When 1 minute of adaptation data was available, and the rank deficiency problem is less severe, BCAT+CSMAPLR performed better than BCAT.

For LF0 features, BCAT+CSMAPLR performed the best as shown in Fig. 6. Although BCAT+CSMAPLR degraded the performance of BCAT in MGC features, for LF0, it performed comparable to SD models after 3 seconds of adaptation data. The number of transformation matrices in LF0 features were much higher compared to MGC as shown in Table 1. This indicates that only a small amount

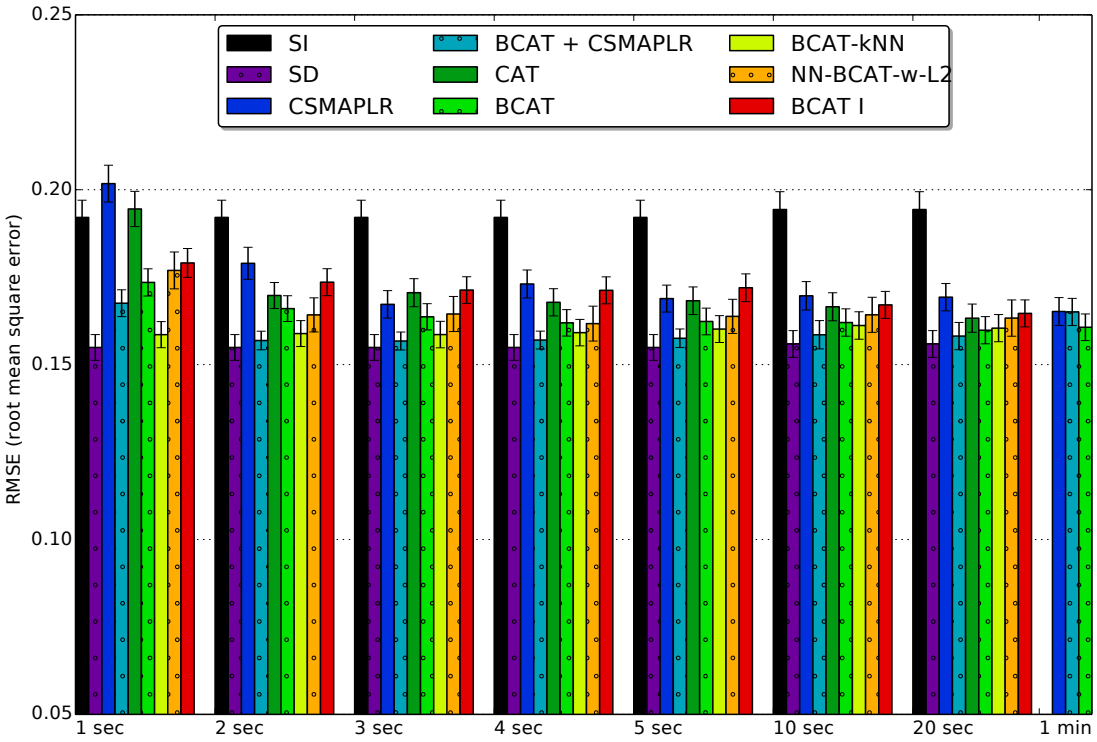

Figure 6: Objective evaluation (RMSE) of adaptation techniques for LF0 features with 95% confidence intervals.

of adaptation data is enough to estimate reliable CSMAPLR transforms for LF0 features. Additionally, this shows that the seed model to CSMAPLR is important and if chosen correctly, output of CSMAPLR can perform comparable with SD with as little as 3 seconds of data. However, for the 1 minute case, performance of CSMAPLR was found to be similar to BCAT+CSMAPLR which indicates that for lf0, performance of the CSMAPLR algorithm does not depend on the seed model when there is sufficient amount of data. Reduced effect of the prior and ability to generate many transformation matrices for lf0 using 1 minute of data increases adaptation flexibility which makes the seed model less important.

BCAT performed significantly better than CAT only in the 1 second case for LF0 features. Because the dimensionality of LF0 supervectors are much lower compared to MGC, BCAT, which relies on the prior cannot significantly outperform CAT for LF0. Similarly, BCAT performs better than BCAT-I only for the 1 sec case.

Even though CSMAPLR does not perform well in one second case for LF0,

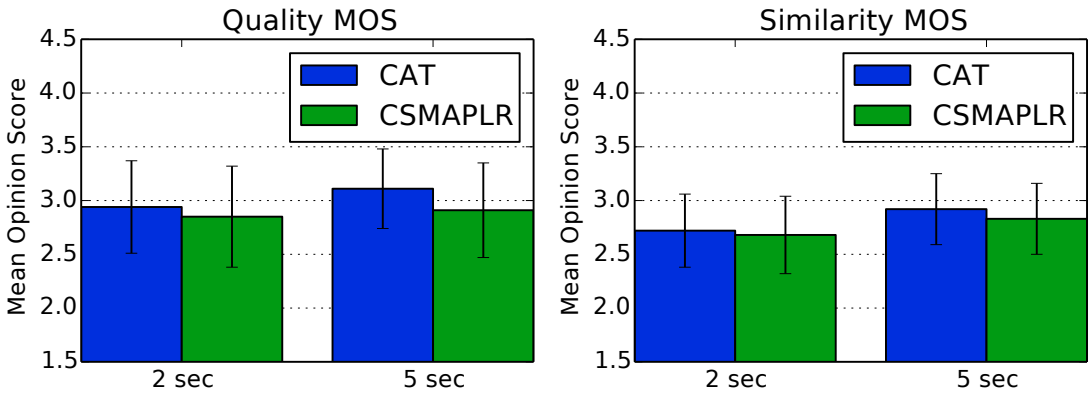

Figure 7: Quality and similarity MOS test results for CSMAPLR and CAT for the 2 and 5 seconds cases.

when more data becomes available, the difference between CSMAPLR and eigenspace algorithms rapidly goes down and at 5 sec case the difference is not significant. BCAT-kNN is significantly better than BCAT for LF0 for the 1 sec and 2 sec cases. The nearest-neighbor selected with BCAT and L2 measure performs as good as BCAT for LF0 in all data sizes.

4.0.13 Subjective Measure Tests

Subjective listening tests are done to evaluate the performance of the proposed systems. ABX tests are used to measure speaker similarity. AB tests are used to measure speech quality. Details of the tests are discussed in Section 4.0.10.

CAT and CSMAPLR algorithms are the baseline systems used in this work. Thus, before comparing their performances with the proposed systems, MOS tests are done to assess the absolute performance of those two systems. Quality and similarity assessment with MOS tests are done for the 2 second and 5 second cases. Performance is not measured for the 1 second case because some of the speakers are not adapted with CSMAPLR for that case due to insufficient amounts of data. 10 listeners took the test. For each case, listeners scored 24 utterances from 6 speakers. Results are shown in Fig. 7. Even though quality of CSMAPLR and CAT algorithms increase slightly from 2 sec to 5 sec, the improvement is not significant. Moreover, difference between the systems are also not significant. In the case of similarity, average performance increases significantly from 2 second to

5 second cases. However, differences between the algorithms are still not significant. Even though there is a large difference between the two systems in objective assessments, differences between them are subtle and cannot be identified in an absolute test such as MOS. However, they were more noticeable in the comparison tests as discussed below.

Subjective tests are done for comparing the perceptual similarity and quality of the proposed algorithms. Results are shown in Table 3 and Table 4. No significant quality or similarity differences between BCAT and BCAT-kNN were found even though their RMSE values are different. Statistical significance is measured using Pearson's chi-squared test. Analyzing the audio, we have found that differences can be heard only in some segments of speech with careful listening. Even though some listeners could hear it sometimes, that was not enough to generate statistically significant differences between the systems. The difference between CAT and proposed algorithms in terms of similarity were also not significant.

BCAT+CSMAPLR algorithm performed worse than BCAT and BCAT-kNN algorithms in terms of similarity for the MGC feature in 2, 3, and 4 seconds cases. Similarly, it was worse than BCAT and BCAT-kNN algorithms in terms of quality for the MGC feature in 3, 4, and 5 sec cases. CSMAPLR was found to distort the models because of rank deficieny when used in tandem with BCAT. This is also reflected in the objective test results as discussed in the previous Section. There was no significant difference between the algorithms for the LF0 case.

Alleviating the perceptual artifacts observed with ML-based CAT was another goal of this work. To measure the improvement in quality with BCAT, AB preference test is used. Results are shown in Fig. 8. Speech quality of BCAT is significantly better than CAT in 1, 2, and, 3 second cases.

Rest of the subjective tests are done to compare CSMAPLR with the proposed algorithms. In each test, the algorithm with the best RMSE performance is compared with CSMAPLR. For the MGC feature, BCAT-kNN has the best RMSE performance over all three systems. For the LF0 feature, BCAT+CSMAPLR

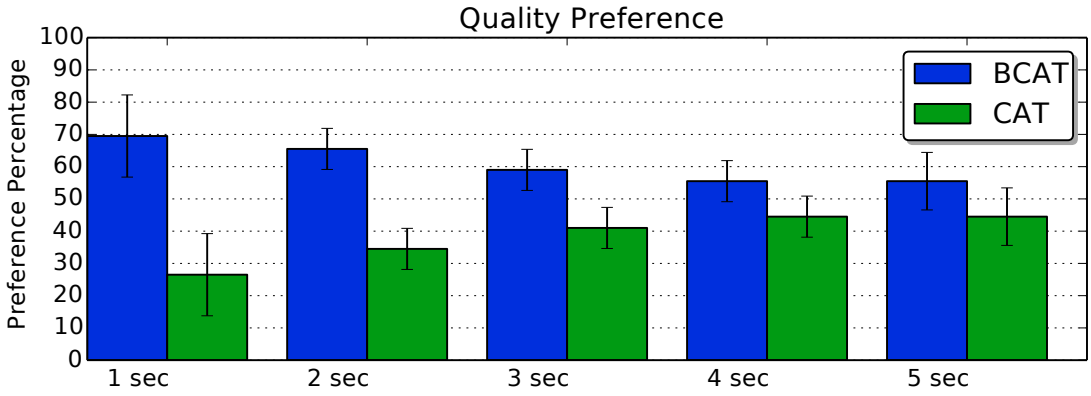

Figure 8: Results of subjective AB preference tests in terms of quality for both MGC and LF0 features with 95% confidence intervals.

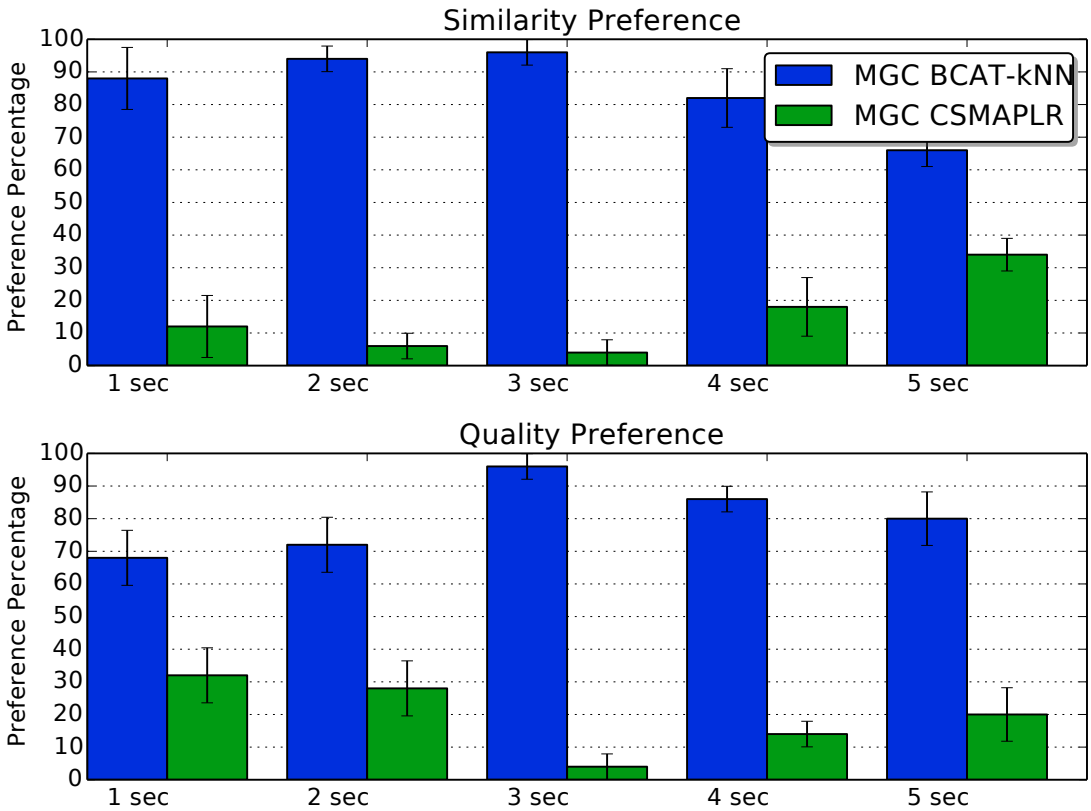

Figure 9: Results of subjective preference tests in terms of similarity (ABX) and quality (AB) for MGC features with 95% confidence intervals. LF0 was fixed to BCAT+CSMAPLR.

Table 3: ABX similarity test results. Statistically significant differences are in bold. MGC and LF0 features are compared seperately. BCAT is used for LF0 when comparing the MGC features. Similarly, BCAT is used for MGC when comparing the LF0 features.

	MGC		LF0	
	BCAT+CSMAPLR	BCAT-kNN	BCAT+CSMAPLR	BCAT-kNN
1 sec	45	55	50	50
2 sec	**32**	**68**	50	50
3 sec	**32**	**68**	45	55
4 sec	**34**	**66**	50	50
5 sec	42.5	57.5	55	45
	BCAT+CSMAPLR	BCAT	BCAT+CSMAPLR	BCAT
1 sec	46	54	54	46
2 sec	**42**	**58**	46	54
3 sec	**40**	**60**	54	46
4 sec	**37.5**	**62.5**	58	42
5 sec	44	56	54	46
	BCAT-kNN	BCAT	BCAT-kNN	BCAT
1 sec	55	45	56	44
2 sec	52.5	47.5	48	52
3 sec	57.5	42.5	52	48
4 sec	52.5	47.5	50	50
5 sec	47.5	52.5	44	56
	BCAT	CAT	BCAT	CAT
1 sec	50	50	57.5	42.5
2 sec	48	52	44	56
3 sec	48	52	57.5	42.5
4 sec	52	48	57.5	42.5
5 sec	56	44	50	50

has the best RMSE performance. In the first listening test, BCAT-kNN is compared with CSMAPLR in terms of MGC features while LF0 features were fixed to BCAT+CSMAPLR in both A and B samples. Results are shown in Fig. 9. BCAT-kNN had a large improvement both in speaker similarity and speech quality over CSMAPLR in MGC adaptations. BCAT-kNN was also compared with BCAT when LF0 is fixed to BCAT+CSMAPLR.

In the second listening test, BCAT+CSMAPLR is compared with CSMAPLR in terms of LF0 while MGC was fixed to BCAT-kNN. Results are shown in Fig. 10. Speaker similarity-wise, BCAT+CSMAPLR had significant improvement

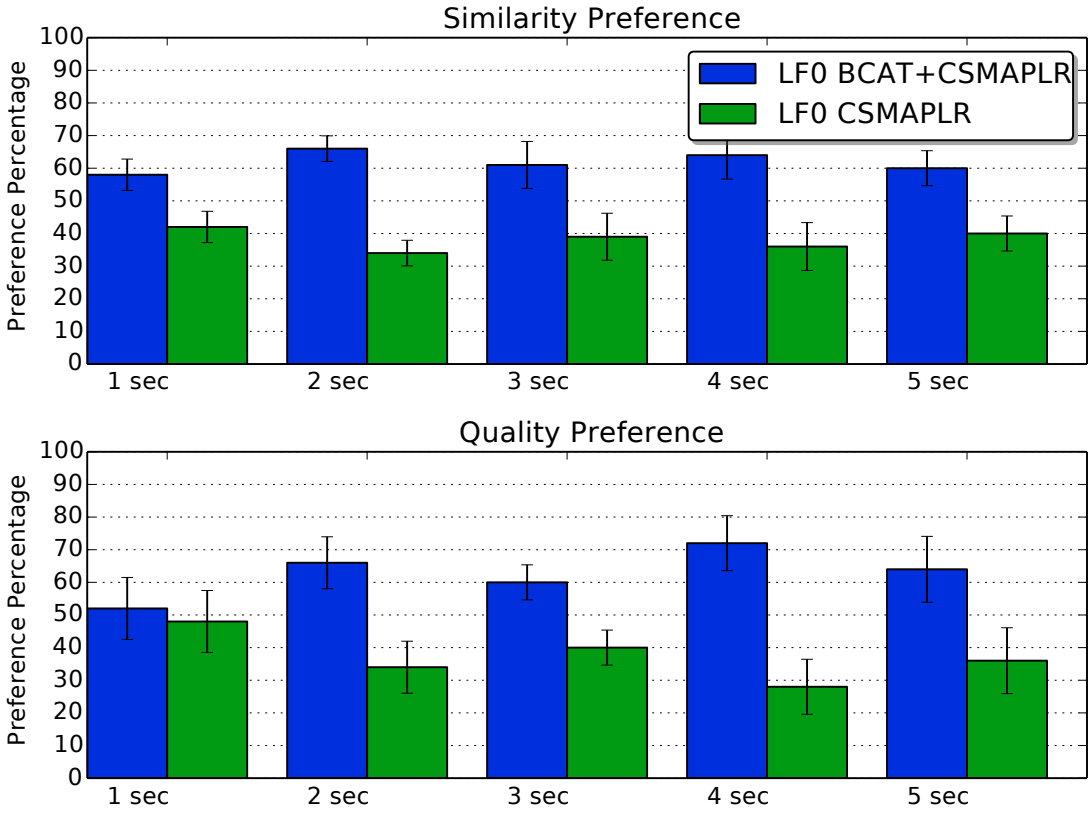

Figure 10: Results of subjective preference tests in terms of similarity (ABX) and quality (AB) for LF0 features with 95% confidence intervals. MGC was fixed to BCAT-kNN.

Table 4: AB quality test results. Statistically significant differences are in bold. MGC and LF0 features are compared seperately. BCAT is used for LF0 when comparing the MGC features. Similarly, BCAT is used for MGC when comparing the LF0 features.

	MGC		LF0	
	BCAT+CSMAPLR	BCAT-kNN	BCAT+CSMAPLR	BCAT-kNN
1 sec	50	50	44	56
2 sec	45	55	48	52
3 sec	**40**	**60**	44	56
4 sec	**40**	**60**	50	50
5 sec	**42**	**58**	50	50
	BCAT+CSMAPLR	BCAT	BCAT+CSMAPLR	BCAT
1 sec	44	56	46	54
2 sec	44	56	46	54
3 sec	**42**	**58**	56	44
4 sec	**40**	**60**	44	56
5 sec	**42**	**58**	56	44
	BCAT-kNN	BCAT	BCAT-kNN	BCAT
1 sec	50	50	48	52
2 sec	52.5	47.5	50	50
3 sec	47.5	52.5	50	50
4 sec	45	55	46	54
5 sec	45	55	50	50

over CSMAPLR. Also, quality-wise, BCAT+CSMAPLR also significant improvement over CSMAPLR except for the 1 second case. These improvements were not as substantial as they were in MGC case.

4.1 Discussion

Even though MOS results of CAT and CSMAPLR algorithms seem similar in Fig. 7, the objective measure tests indicate significant differences between the two algorithms. Moreover, CAT was found to outperform CSMAPLR in the literature when tiny amounts of adaptation data is available [15]. To compare them in more detail, ABX similarity test and AB quality test are performed. Results are shown in Fig. 11. We have found that the listeners can hear the differences between the two algorithms better in comparison tests as opposed to MOS tests. Quality-wise CAT produced less annoying artifacts than CSMAPLR. Moreover, listeners commented that overall quality was sometimes better with CAT even when there

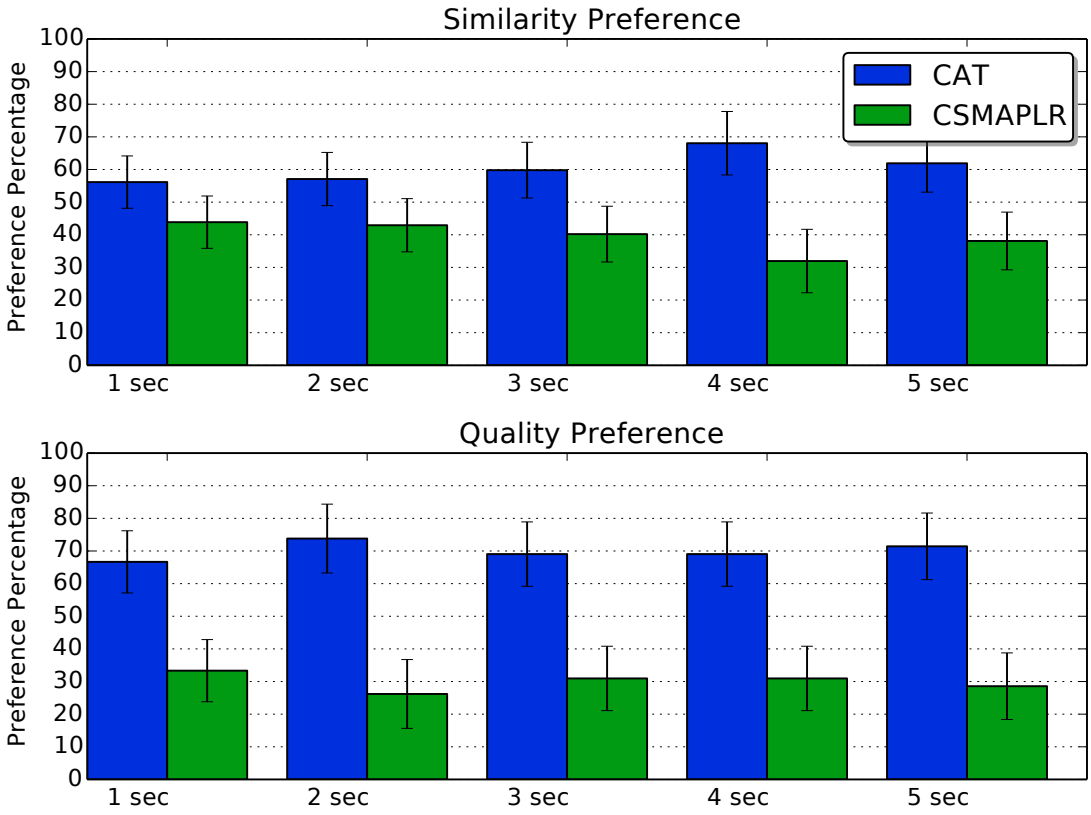

Figure 11: Results of subjective preference tests in terms of similarity (ABX) and quality (AB) for both MGC and LF0 features with 95% confidence intervals.

are no artifacts.

The goal of the BCAT algorithm was to reduce the perceptual artifacts of CAT that occur due to overfitting. We have found that regularization through using a prior distribution is helpful for removing those artifacts. The improvements were observed both in objective and subjective quality tests. However, similarity-wise, significant differences were not observed. Thus, even though some of the states were distorted with CAT, overall similarity to the target speaker was judged to be similar to BCAT.

BCAT-kNN algorithm was designed to exploit the nearest-neighbors to the target speaker for learning the eigenspace and computing the covariance matrix of the prior distribution. Even though it performed well especially for the MGC features in the objective tests, it was not better than BCAT in the listening tests. We have found that the improvements in MGC with the BCAT-kNN algorithm is mostly related to removal of outliers from the training data. Thus, our results indicate that those outliers do not cause significant perceptual distortions in synthesis.

We hypothesized that using the CSMAPLR algorithm in tandem with the BCAT algorithm can help improve the performance of BCAT. However, the CSMAPLR algorithm sometimes created artifacts in the MGC features because of the rank deficiency problem which lowered the quality and similarity of speech. Only when one minute of adaptation data was available, improvements in the MGC features were observed with the tandem approach. Even though significant performance improvements were possible for lf0 features in the objective tests, those improvements were mostly not noticed by the listeners in the listening tests. Variations of lf0 parameter is significantly reduced by the parameter generation algorithm due to smoothing which made it harder for the listeners to hear differences.

CHAPTER V

CONCLUSION

In this work, we proposed three algorithms for rapid speaker adaptation of SSS models with few seconds of adaptation data. First method is a Bayesian eigenvoice, BCAT, approach to rapid speaker adaptation when feature dimensionality is high and data is scarce as in the case of MGC features. We have shown how eigenvoice based adaptations can be improved by adding prior information regarding the weight vectors to constrain the estimation.

In the second method, BCAT-kNN, the eigenspace and the weight vector priors are estimated using only k target-specific nearest neighbors to constrain the estimation further.

The third algorithm is a tandem BCAT/CSMAPLR approach for LF0 features where dimensionality is low. BCAT adaptation can only have a coarse estimation of the target model since it works in a limited space. Also, its performance saturates quickly when more data becomes available. Because of those, we proposed another iteration of adaptation step after BCAT using CSMAPLR so that adaptation can continue to improve with more data. In this approach, output of the BCAT algorithm is used as the seed model for CSMAPLR adaptation instead of the original SI model.

Both for LF0 and MGC, significant improvements in objective tests are achieved compared to baseline CSMAPLR and CAT algorithms. Similarly, in subjective listening tests, proposed algorithms outperformed CSMAPLR algorithm in both quality and similarity. Moreover, quality is improved using the proposed methods compared to CAT algorithm. However, similarity-wise, differences between CAT and the three proposed methods are subtle and could not be consistently identified by the listeners.

Success of the BCAT-kNN algorithm depends on selecting good NN's. Therefore, several distance measures are compared to select the best nearest-neighbor. L2 distance of $\boldsymbol{w}$ vectors of BCAT adaptations were found to have the best performance both for MGC and LF0 features. In our future work, we will explore new distance measures to select better nearest neighbors in order to further improve the performance of the proposed algorithms.

Bibliography

[1] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," 2013.

[2] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing: a guide to theory, algorithm, and system development*. Upper Saddle River, NJ: Prentice Hall PTR, 2001.

[3] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, p. 10391064, 2009.

[4] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE*, vol. E90-D, Feb. 2007.

[5] J. Yamagishi, B. Usabaev, S. King, O. Watts, J. Dines, J. Tian, Y. Guan, R. Hu, K. Oura, Y. Wu, *et al.*, "Thousands of voices for HMM-based speech synthesis-analysis and application of TTS systems built on various ASR corpora," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 984–1004, 2010.

[6] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 66–83, 2009.

[7] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 357–366, 1995.

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

[9] C. Breslin, K. K. Chin, M. J. Gales, K. Knill, and H. Xu, "Prior information for rapid speaker adaptation.," in *INTERSPEECH*, pp. 1644–1647, 2010.

[10] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, pp. 695 –707, nov 2000.

[11] 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.

[12] M. J. F. Gales, "Cluster adaptive training of hidden Markov models," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 4, pp. 417–428, 2000.

[13] H. Zen, N. Braunschweiler, S. Buchholz, M. J. Gales, K. Knill, S. Krstulovic, and J. 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.

[14] J. Latorre, V. Wan, M. J. Gales, L. Chen, K. K. Chin, K. Knill, and M. Akamine, "Speech factorization for HMM-TTS based on cluster adaptive training.," in *INTERSPEECH*, 2012.

[15] V. Wan, J. Latorre, K. K. Chin, L. Chen, M. J. Gales, H. Zen, K. Knill, and M. Akamine, "Combining multiple high quality corpora for improving HMM-TTS.," in *INTERSPEECH*, 2012.

[16] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: a maximum likelihood approach to speaker normalization," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2, pp. 1043–1046, apr 1997.

[17] M. Charlier, Y. Ohtani, T. Toda, A. Moinet, and T. Dutoit, "Cross-language voice conversion based on eigenvoices," in *INTERSPEECH*, pp. 1635–1638, 2009.

[18] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Speaker adaptive training for one-to-many eigenvoice conversion based on gaussian mixture model.," in *INTERSPEECH*, pp. 1981–1984, ISCA, 2007.

[19] T. Yoshimura, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[20] J. Yamagishi, T. Masuko, and T. Kobayashi, "HMM-based expressive speech synthesis-Towards TTS with arbitrary speaking styles and emotions," in *Proc. of Special Workshop in Maui (SWIM)*, 2004.

[21] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[22] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer speech and language*, vol. 9, no. 2, p. 171, 1995.

[23] O. Siohan, T. A. Myrvoll, and C.-H. Lee, "Structural maximum a posteriori linear regression for fast HMM adaptation," *Computer Speech & Language*, vol. 16, no. 1, pp. 5–24, 2002.

[24] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 788–798, may 2011.

[25] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 49–60, jan 1998.

[26] D. Y. Kim, S. Umesh, M. J. F. Gales, T. Hain, and P. C. Woodland, "Using VTLN for broadcast news transcription," in *Proc. ICSLP*, vol. 4, 2004.

[27] P. Akhil, S. P. Rath, S. Umesh, and D. R. Sanand, "A computationally efficient approach to warp factor estimation in VTLN using EM algorithm and sufficient statistics.," in *INTERSPEECH*, pp. 1713–1716, 2008.

[28] M. J. Gales and R. C. van Dalen, "Predictive linear transforms for noise robust speech recognition," in *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pp. 59–64, IEEE, 2007.

[29] F. Flego and M. J. Gales, "Incremental predictive and adaptive noise compensation," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 3837–3840, IEEE, 2009.

[30] L. Saheer, J. Yamagishi, P. Garner, and J. Dines, "Combining vocal tract length normalization with hierarchical linear transformations," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 8, pp. 262–272, April 2014.

[31] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Eurospeech*, pp. 2347–2350, 1999.

[32] L. Chen, M. J. Gales, V. Wan, J. Latorre, and M. Akamine, "Exploring rich expressive information from audiobook data using cluster adaptive training.," in *INTERSPEECH*, 2012.

[33] D. Tani, T. Toda, Y. Ohtani, H. Saruwatari, and K. Shikano, "Maximum a posteriori adaptation for many-to-one eigenvoice conversion," in *Proc. INTERSPEECH*, pp. 1461–1464, 2008.

[34] Y. Wu and R. Wang, "Minimum generation error training for HMM-based speech synthesis," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, 2006.

[35] "HMM-based speech synthesis system (HTS). [online]. available: http://hts.sp. nitech.ac.jp/."

[36] M. Gibson and W. Byrne, "Unsupervised intralingual and cross-lingual speaker adaptation for HMM-based speech synthesis using two-pass decision tree construction," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 895–904, 2011.

[37] X. Peng, K. Oura, Y. Nankaku, and K. Tokuda, "Cross-lingual speaker adaptation for HMM-based speech synthesis considering differences between language-dependent average voices," in *Proc. ICSP*, pp. 605–608, 2010.

[38] Y. J. Wu, S. King, and K. Tokuda, "Cross-lingual speaker adaptation for HMM-based speech synthesis," in *Chinese Spoken Language Processing, 2008. ISCSLP'08. 6th International Symposium on*, pp. 1–4, 2008.

[39] M. Tachibana, S. Izawa, T. Nose, and T. Kobayashi, "Speaker and style adaptation using average voice model for style control in HMM-based speech synthesis," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 4633–4636, 2008.

[40] H. Liang, J. Dines, and L. Saheer, "A comparison of supervised and unsupervised cross-lingual speaker adaptation approaches for HMM-based speech synthesis," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 4598–4601, 2010.

[41] K. Oura, K. Tokuda, J. Yamagishi, S. King, and M. Wester, "Unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 4594–4597, 2010.

[42] O. Watts, J. Yamagishi, S. King, and K. Berkling, "Synthesis of child speech with HMM adaptation and voice conversion," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 1005–1016, 2010.

[43] R. Karhila, D. R. Sanand, M. Kurimo, and P. Smit, "Creating synthetic voices for children by adapting adult average voice using stacked transformations and VTLN," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4501–4504, 2012.

[44] L. Saheer, J. Dines, and P. N. Garner, "Vocal tract length normalization for statistical parametric speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 7, pp. 2134–2148, 2012.

[45] J. Yamagishi, T. Nose, H. Zen, Z. H. Ling, T. Toda, K. Tokuda, S. King, and S. Renals, "Robust speaker-adaptive HMM-based text-to-speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 1208–1230, 2009.

[46] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 2, pp. 805–808, 2001.

[47] T. Nose, Y. Kato, and T. Kobayashi, "A speaker adaptation technique for MRHSMM-based style control of synthetic speech," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. IV–833, 2007.

[48] J. Yamagishi, T. Kobayashi, M. Tachibana, K. Ogata, and Y. Nakano, "Model adaptation approach to speech synthesis with diverse voices and styles," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. IV–1233, 2007.

[49] S. Andraszewicz, J. Yamagishi, and S. King, "Vocal attractiveness of statistical speech synthesisers," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5368–5371, 2011.

[50] J. Yamagishi and T. Kobayashi, "Adaptive training for hidden semi-markov model," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 365–368, 2005.

[51] L. Saheer, J. Yamagishi, P. N. Garner, and J. Dines, "Combining vocal tract length normalization with hierarchial linear transformations," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4493–4496, 2012.

[52] R. Karhila, D. R. Sanand, M. Kurimo, and P. Smit, "Creating synthetic voices for children by adapting adult average voice using stacked transformations and VTLN," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4501–4504, 2012.

[53] C. P. Chen, Y. C. Huang, C. H. Wu, and K. D. Lee, "Cross-lingual frame selection method for polyglot speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4521–4524, 2012.

[54] Y. Shiga, "Effect of anti-aliasing filtering on the quality of speech from an HMM-based synthesizer," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4525–4528, 2012.

[55] O. Watts, "Unsupervised learning for text-to-speech synthesis,"

[56] Z. H. Ling, K. Richmond, and J. Yamagishi, "Vowel creation by articulatory control in HMM-based parametric speech synthesis," *The Listening Talker*, p. 72, 2012.

[57] R. Dall, C. Veaux, J. Yamagishi, and S. King, "Analysis of speaker clustering strategies for HMM-based speech synthesis,"

[58] S. Takamichi, T. Toda, Y. Shiga, H. Kawai, S. Sakti, and S. Nakamura, "An evaluation of parameter generation methods with rich context models in HMM-Based speech synthesis,"

[59] H. Lu and S. King, "Using bayesian networks to find relevant context features for HMM-based speech synthesis,"

[60] V. Chunwijitra, T. Nose, and T. Kobayashi, "A speech parameter generation algorithm using local variance for HMM-based speech synthesis," in *Proc. 13th Annual Conference of the International Speech Communication Association*, 2012.

[61] T. Raitio, A. Suni, M. Vainio, and P. Alku, "Wideband parametric speech synthesis using warped linear prediction,"

[62] H. Siln, E. Helander, J. Nurminen, and M. Gabbouj, "Ways to implement global variance in statistical speech synthesis,"

[63] B. Bollepalli, A. W. Black, and K. Prahallad, "Modeling a noisy-channel for voice conversion using articulatory features,"

[64] A. C. Janska, E. Schrger, T. Jacobsen, and R. A. J. Clark, "Asymmetries in the perception of synthesized speech,"

[65] E. Greene, T. Mishra, P. Haffner, and A. Conkie, "Predicting character-appropriate voices for a TTS-based storyteller system,"

[66] A. Sorin, S. Shechtman, and V. Pollet, "Psychoacoustic segment scoring for multi-form speech synthesis,"

[67] R. Gomez, T. Toda, H. Saruwatari, and K. Shikano, "Techniques in rapid unsupervised speaker adaptation based on HMM-Sufficient statistics," *Speech Communication*, vol. 51, no. 1, pp. 42–57, 2009.

[68] {K. Shichiri}, {A. Sawabe}, {K. Tokuda}, {T. Masuko}, {T. Kobayashi}, and {T. Kitamura}, "Eigenvoices for HMM-based speech synthesis," *Proc. of ICSLP*, pp. pp.1269–1272, Sept. 2002.

[69] C. Breslin, K. K. Chin, M. J. Gales, and K. Knill, "Integrated online speaker clustering and adaptation.," in *INTERSPEECH*, pp. 1085–1088, 2011.

[70] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice conversion based on gaussian mixture model," in *Ninth International Conference on Spoken Language Processing*, 2006.

[71] T. Masuko, T. Kobayashi, and K. Miyanaga, "A style control technique for HMM-based speech synthesis," in *Eighth International Conference on Spoken Language Processing*, 2004.

[72] K. Kazumi, Y. Nankaku, and K. Tokuda, "Factor analyzed voice models for HMM-based speech synthesis," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 4234–4237, 2010.

[73] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Mixed excitation for HMM-based speech synthesis," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[74] K. Shinoda and C.-H. Lee, "A structural bayes approach to speaker adaptation," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, pp. 276 –287, mar 2001.

[75] M. Gales, "Cluster adaptive training of hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 417–428, 1999.

[76] T. Tomoki and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE transactions on information and systems*, vol. 90, no. 5, p. 816, 2007.

[77] Y. Qian, F. Soong, Y. Chen, and M. Chu, "An HMM-based Mandarin Chinese text-to-speech system," *Chinese Spoken Language Processing*, pp. 223–232, 2006.

[78] J. Yamagishi and S. King, "Simple methods for improving speaker-similarity of HMM-based speech synthesis," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 4610–4613, 2010.

[79] S. Tiomkin, D. Malah, S. Shechtman, and Z. Kons, "A hybrid text-to-speech system that combines concatenative and statistical synthesis units," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. pp, no. 99, 2010.

[80] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptation for HMM-based speech synthesis system using MLLR," in *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.

[81] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3, pp. 1315–1318, IEEE, 2002.

[82] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis.," *IEICE Transactions on Information and Systems*, vol. 83, no. 11, pp. 2099–2107, 2000.

[83] A. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, IEEE, 2007.

[84] O. Abdel-Hamid, S. Abdou, and M. Rashwan, "Improving Arabic HMM based speech synthesis quality," in *Ninth International Conference on Spoken Language Processing*, ISCA, 2006.

[85] X. Gonzalvo, I. Iriondo, J. Socoró, F. Alıas, and C. Monzo, "HMM-based Spanish speech synthesis using CBR as F0 estimator," *ITRW on NOLISP*, 2007.

[86] K. Oflazer and S. Inkelas, "A finite state pronunciation lexicon for Turkish," in *Proceedings of the EACL Workshop on Finite State Methods in NLP, Budapest, Hungary*, vol. 82, pp. 900–918, Citeseer, 2003.

[87] I. Ergenc, *Spoken Language And Dictionary Of Turkish Articulation*. MULTILINGUAL, 2002.

[88] C. Benoīt, M. Grice, and V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences," *Speech Communication*, vol. 18, no. 4, pp. 381–392, 1996.

[89] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of Sixth ISCA Workshop on Speech Synthesis*, pp. 294–299, Citeseer, 2007.

[90] K. Tokuda, H. Zen, J. Yamagishi, T. Masuko, S. Sako, A. Black, and T. Nose, "The hmm-based speech synthesis system (hts) version 2.2," [Online].

[91] S. Kinga and V. Karaiskosb, "The blizzard challenge 2009,"

[92] H. Zen, "Speaker and language adaptive training for HMM-based polyglot speech synthesis," in *INTERSPEECH*, pp. 410–413, 2010.

[93] P. Taylor, A. Black, and R. Caley, "The architecture of the Festival speech synthesis system," in *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, Citeseer, 1998.

VITA

Amir Mohammadi graduated from Electrical/Biomedical Engineering (bio-electric) program of University of Tehran in 2011. In 2011, he started Msc. in Electrical Engineering at Özyeğin University. He started working on Speech Processing at the Speech Processing Lab. He has been working on Speaker Adaptation Algorithms in TTS, Spoofing Automatic Speaker Verification Systems and their Countermeasures, Feature Selection and Speaker Visualization, and Hybrid Speech Synthesis Algorithms. He published 1 journal paper in IEEE/ACM transactions on Audio, Speech, and Language Processing and 4 conference papers during his master studies.