ISCA Archive
http://www.isca-speech.org/archive
8th ISCA Speech Synthesis Workshop • August 31 – September 2, 2013 • Barcelona, Spain

Noise Robustness in HMM-TTS Speaker Adaptation

Kayoko Yanagisawa, Javier Latorre, Vincent Wan, Mark J. F. Gales and Simon King

Toshiba Research Europe Ltd., Cambridge Research Lab, 208 Science Park, Cambridge, UK
{kayoko.yanagisawa,javier.latorre,vincent.wan,mjfg}@crl.toshiba.co.uk,Simon.King@ed.ac.uk

Abstract

Speaker adaptation for TTS applications has been receiving more attention in recent years for applications such as voice customisation or voice banking. If these applications are offered as an Internet service, there is no control on the quality of the data that can be collected. It can be noisy with people talking in the background or recorded in a reverberant environment. This makes the adaptation more difficult. This paper explores the effect of different levels of additive and convolutional noise on speaker adaptation techniques based on cluster adaptive training (CAT) and average voice model (AVM). The results indicate that although both techniques suffer degradation to some extent, CAT is in general more robust than AVM.

Index Terms: speech synthesis, cluster adaptive training, speaker adaptation, average voice models, noise robust adaptation

1. Introduction

With the arrival of smartphones and tablets, text-to-speech (TTS) systems are becoming more and more ubiquitous. However, most are still limited in the number of voices and/or expressions they can provide. For users of TTS applications such as Augmentative and Alternative Communication (AAC) devices, the ability to be uniquely identified by their own voice is an important aspect. For more casual users of TTS, too, personalisation can add value to the TTS system.

Building a good quality speaker-dependent TTS voice requires a large database of recordings with good phonetic coverage made in a controlled environment. This is not a realistic scenario for personalisation with dysarthric patients or with a casual user of TTS. Speaker adaptation techniques for Hidden Markov model based TTS (HMM-TTS) have emerged in recent years, which allow the creation of a target speaker's voice using a small amount of speech [1]. An average voice model (AVM) is trained on a large corpus containing multiple speakers. It is then adapted to a voice using the target speaker's adaptation data. They found that six minutes of adaptation data was enough to build a voice that sounds more natural than that of a speaker-dependent system trained on thirty minutes of speech.

The ability to create one's own voice with a small amount of data opens up the possibility to offer voice personalisation capabilities to a wider public. For example, an online custom voice building service could be envisaged where users submit a small number of sentences recorded at home, which are used to adapt a pre-built model to create their voice. In that scenario, the adaptation data is likely to be recorded in an uncontrolled environment, so the data might contain background noise, reverberation, different channel effects due to the use of non-professional recording equipment, and/or various signal processing applied by the sound card. Each of these factors has a strong impact on the quality of the models that can be obtained. In order to deal with these problems, robustness to noise is a requirement for speaker adaptation systems.

Noise robustness is a well known topic in the field of automatic speech recognition (ASR) but relatively new for TTS. The effect of creating AVMs from 'noisy' ASR data was investigated in [2, 3]. The ultimate goal of that work was to produce models that could be shared by the ASR and the TTS engine of a speech-to-speech translation system. For that purpose, the effect of training TTS models on noisy ASR data was investigated. The results showed degradation with respect to training models on clean speech. However, TTS and ASR models do not need to be shared in most cases, which means that TTS systems can be trained on reasonably good data. However, the problem regarding the quality of adaptation data remains.

Some of the noise in the adaptation data can be reduced by signal processing. For example, pops can be reduced with a high-pass filter and background noise can be reduced using spectral subtraction. More sophisticated techniques were applied in [4]. These techniques can be expected to improve the adaptation outcome but there is a limit to the amount and type of noise that can be removed. This poses a problem for AVMs because the strong adaptation capability of CMLLR transforms will treat the remaining noise as part of the speaker's voice.

Multiple linear regressions systems, such as Cluster Adaptive Training (CAT) [5], Multiple-regression HSMM (MRHSMM) [6] or eigenvoices [7] also allow speaker adaptation. In these techniques, adaptation data is projected into a linear space trained on clean data. The idea is similar to the signal-subspace approach proposed for speech enhancement [8]. As the number of parameters needed by these systems is much smaller than for AVM, their adaptation capability is much weaker, especially with large amounts of adaptation data. However, this also makes them more robust with sparse adaptation data. Moreover, the speaker space corresponds to clean speech so they may be more robust to noise.

This paper studies the effect of adapting AVM and CAT models to data with different levels of background noise and reverberation. Section 2 reviews and compares CAT and AVM adaptation. Section 3 describes the experiments. Section 4 analyses speaker similarity. Section 5 concludes.

2. Cluster adaptive training

The main characteristic of CAT [9] is that the means of the distributions are linear combinations of the mean vectors of two or more clusters. In such a model, the emission probability of an observation vector for a given speaker s, and component m is

$$p(\boldsymbol{o}(t) \mid m, s, \mathcal{M}) = \mathcal{N}\left(\boldsymbol{o}(t); \boldsymbol{\mu}_m^{(s)}, \boldsymbol{\Sigma}_{v(m)}\right) \quad (1)$$

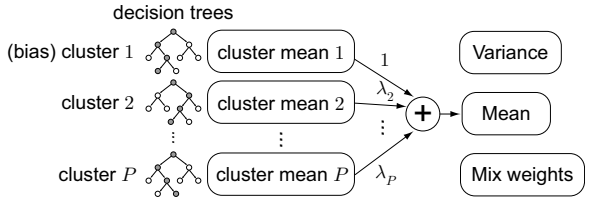


Figure 1: CAT with cluster-dependent decision trees.

with

$$\boldsymbol{\mu}_m^{(s)} = \boldsymbol{\mu}_{c(m,1)} + \boldsymbol{M}_m \boldsymbol{\lambda}_{q(m)}^{(s)} \tag{2}$$

$$\boldsymbol{\lambda}_{q(m)}^{(s)} = \left[\lambda_{2,q(m)}^{(s)}, \ldots, \lambda_{P,q(m)}^{(s)}\right]^\top \tag{3}$$

$$\boldsymbol{M}_m = \left[\boldsymbol{\mu}_{c(m,2)}, \ldots, \boldsymbol{\mu}_{c(m,P)}\right] \tag{4}$$

where $t \in \{1, \ldots, T\}$, $m \in \{1, \ldots, M\}$ and $s \in \{1, \ldots, S\}$ enumerate the frames, Gaussian components and speakers respectively; $q(m) \in \{1, \ldots, Q\}$ and $v(m) \in \{1, \ldots, V\}$ are respectively the m^{th} component's CAT regression classes and leaf node in the covariance matrices' decision tree; $c(m,i) \in \{1, \ldots, N\}$ is the leaf node for cluster i of component m in decision trees for cluster mean vectors; P is the number of clusters; $\boldsymbol{o}(t)$ is the observation vector at frame t; $\lambda_{i,q}^{(s)}$ and $\boldsymbol{\lambda}_q^{(s)}$ are respectively the i^{th} cluster's CAT weight and the weight vectors for speaker s associated with CAT regression class q; $\boldsymbol{\mu}_n$ is the cluster mean vector associated with leaf node n; $\boldsymbol{M}_m$ is component m's matrix of cluster mean vectors; $\boldsymbol{\Sigma}_k$ is leaf node k's covariance matrix; $\mathcal{M}$ is the full set of model parameters.

When each cluster is allowed its own decision tree, the result is a multi-tree model with tree-intersection as depicted in Figure 1. The main advantage of this model is its capacity to model a large number of contexts with a reduced number of parameters. An important characteristic is that the weight of the first (bias) cluster is always 1. The reason is that the goal of the bias cluster is to model those attributes which are common to all speakers. Covariance matrices and priors for the multi-space distributions (MSD) [10] could have their own tying structures, but usually they share the decision trees of the bias cluster.

The auxiliary function of the EM algorithm for the distribution of (1) is

$$\mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}}) = -\frac{1}{2} \sum_{m,t,s} \gamma_m(t,s) \times \left\{ \left(\boldsymbol{o}(t) - \boldsymbol{\mu}_m^{(s)}\right)^\top \boldsymbol{\Sigma}_{v(m)}^{-1} \left(\boldsymbol{o}(t) - \boldsymbol{\mu}_m^{(s)}\right) + \log\left|\boldsymbol{\Sigma}_{v(m)}\right| \right\} + C \tag{5}$$

where C is a constant, $\hat{\mathcal{M}}$ is the current estimate of $\mathcal{M}$, and $\gamma_m(t,s)$ is the posterior probability of component m generating $\boldsymbol{o}(t)$ given s and $\hat{\mathcal{M}}$. Maximising $\mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}})$ w.r.t. $\boldsymbol{\mu}_n$ yields

$$\hat{\boldsymbol{\mu}}_n = \boldsymbol{G}_{nn}^{-1} \left(\boldsymbol{k}_n - \sum_{\nu \neq n} \boldsymbol{G}_{n\nu} \boldsymbol{\mu}_\nu \right) \tag{6}$$

where

$$\boldsymbol{G}_{n\nu} = \sum_{\substack{m,i,j \\ c(m,i)=n \\ c(m,j)=\nu}} \boldsymbol{G}_{ij}^{(m)}, \qquad \boldsymbol{k}_n = \sum_{\substack{m,i \\ c(m,i)=n}} \boldsymbol{k}_i^{(m)} \tag{7}$$

and $\boldsymbol{G}_{ij}^{(m)}$ and $\boldsymbol{k}_i^{(m)}$ are accumulated statistics defined as

$$\boldsymbol{G}_{ij}^{(m)} = \sum_{t,s} \gamma_m(t,s) \lambda_{i,q(m)}^{(s)} \boldsymbol{\Sigma}_{v(m)}^{-1} \lambda_{j,q(m)}^{(s)} \tag{8}$$

$$\boldsymbol{k}_i^{(m)} = \sum_{t,s} \gamma_m(t,s) \lambda_{i,q(m)}^{(s)} \boldsymbol{\Sigma}_{v(m)}^{-1} \boldsymbol{o}(t). \tag{9}$$

By combining (6) for all the mean vectors, the update equations can be written as

$$\begin{bmatrix} \boldsymbol{G}_{11} & \ldots & \boldsymbol{G}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{G}_{N1} & \ldots & \boldsymbol{G}_{NN} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}}_1 \\ \vdots \\ \hat{\boldsymbol{\mu}}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}_1 \\ \vdots \\ \boldsymbol{k}_N \end{bmatrix}. \tag{10}$$

The order of (10) can be very large but it is a sparse optimisation problem because for most leaves $\boldsymbol{G}_{n,\nu} = \boldsymbol{0}$. Furthermore, if the covariances are diagonal, each dimension can be solved independently. The update equations for $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\lambda}_q^{(s)}$ are

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_{\substack{t,s,m \\ v(m)=k}} \gamma_m(t,s) \left(\boldsymbol{o}(t) - \boldsymbol{\mu}_m^{(s)}\right) \left(\boldsymbol{o}(t) - \boldsymbol{\mu}_m^{(s)}\right)^\top}{\sum_{\substack{t,s,m \\ v(m)=k}} \gamma_m(t,s)} \tag{11}$$

$$\boldsymbol{\lambda}_q^{(s)} = \Big(\sum_{\substack{t,m \\ q(m)=q}} \gamma_m(t,s) \boldsymbol{M}_m^\top \boldsymbol{\Sigma}_{v(m)}^{-1} \boldsymbol{M}_m \Big)^{-1} \sum_{\substack{t,m \\ q(m)=q}} \gamma_m(t,s) \boldsymbol{M}_m^\top \boldsymbol{\Sigma}_{v(m)}^{-1} \boldsymbol{o}(t). \tag{12}$$

2.1. Tree building

Tree building in a tree-intersection model is computationally expensive [11]. To solve this, a cluster by cluster approach is used [12] in which the tree for one cluster is updated while the trees of the other clusters and their canonical parameters are held fixed. As usual, each tree is built to maximise the log-likelihood given the training data. Following [13], the log-likelihood for the n^{th} node in the i^{th} cluster can be computed as

$$\mathcal{L}(n) = \frac{1}{2} \hat{\boldsymbol{\mu}}_n^\top \left(\sum_{m \in \mathcal{S}(n)} \boldsymbol{G}_{ii}^{(m)} \right) \hat{\boldsymbol{\mu}}_n \tag{13}$$

with $\hat{\boldsymbol{\mu}}_n$ the ML estimate of $\boldsymbol{\mu}_n$ which is

$$\hat{\boldsymbol{\mu}}_n = \left(\sum_{m \in \mathcal{S}(n)} \boldsymbol{G}_{ii}^{(m)} \right)^{-1} \times \sum_{m \in \mathcal{S}(n)} \left(\boldsymbol{k}_i^{(m)} - \sum_{j \neq i} \boldsymbol{G}_{ij}^{(m)} \boldsymbol{\mu}_{c(m,j)} \right). \tag{14}$$

For each node n, the optimum split question q is the one that maximises the log-likelihood gain

$$\mathcal{L}(n; q) = \mathcal{L}(n_+^q) + \mathcal{L}(n_-^q) - \mathcal{L}(n). \tag{15}$$

In this way, the best question to split the n^{th} node can be selected based on the log-likelihood gain. The splitting process is stopped when a reasonable balance between complexity and accuracy is achieved. In the experiments in Section 3, minimum description length (MDL) [14] was used. After constructing the decision trees for a cluster, decision trees for the next cluster are re-built in the same manner. This process is repeated from cluster 1 to P, and the whole process repeated as desired.

2.2. CAT vs. AVM

In CAT, speaker variety is captured by the weight vector $\boldsymbol{\lambda}^{(s)}$. This vector may be interpreted as a point in eigenspace representing all possible speakers. The space is spanned by the bases defined by the CAT clusters. Since the CAT model is trained on clean speech, this speaker space is expected to be clean also. Given that CAT adaptation estimates only the $\boldsymbol{\lambda}^{(s)}$, there are insufficient degrees of freedom to capture noise in the adaptation data. Therefore CAT adaptation is constrained to yield (mostly) clean synthesis.

In contrast, the emission probability for a given component and speaker using CMLLR or CSMAPLR transforms is

$$p(\boldsymbol{o}(t) \mid m, s, \mathcal{M}) = |\boldsymbol{A}^{(s)}_{r(m)}| \mathcal{N}\left(\boldsymbol{A}^{(s)}_{r(m)}\boldsymbol{o}(t) + \boldsymbol{b}^{(s)}_{r(m)}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right) \tag{16}$$

where $r(m) \in \{1, \ldots, R\}$ is the regression class associated with component m and $\{\boldsymbol{A}^{(s)}_{r(m)}, \boldsymbol{b}^{(s)}_{r(m)}\}$ the CMLLR transform associated with class $r(m)$. Comparing (16) and (1) it is obvious that a CMLLR transform is much more powerful than the CAT weights, allowing not just translations but also rotations of the acoustic space. This allows AVM to produce better speaker similarity than CAT when there is sufficient adaptation data. However, when the adaptation data is sparse, some of the transforms can not be estimated robustly. Therefore, both the similarity and the quality degrades [15]. In the case of adaptation with noisy data, this extra freedom might also be problematic. CMLLR has no mechanism to constrain the adapted models within a sub-space of "clean" speech. Thus it is likely to treat the noise as an attribute of the speaker.

3. Experiments

3.1. Data

Speech data from a variety of sources were used for training the models. They consisted of a) high quality recordings of phonetically balanced sentences read by professional voice talents with a neutral style in specialist recording studios; b) cheaper, lower quality studio recordings made in less strictly controlled conditions; and c) amateur-read audiobooks which were published freely on the Internet. In total, 20 speakers provided just under 30 hours of data and they all spoke US English with the General American accent.

A separate test set consisting of 8 male and 8 female non-professional speakers was recorded for adaptation. The recordings were of neutrally read sentences made in quiet office rooms using a headset microphone on a laptop with all signal processing effects turned off. Each speaker spoke the same set of 100 sentences, amounting to about 7 minutes of speech per speaker. All speakers spoke US English but not strictly the General American accent.

3.1.1. Simulation of noise

The clean data was corrupted with additive and convolutional noise to simulate noisy adaptation data. Multi-speaker babble consisting of real world multi-talker non-stationary environment noise captured at a trade show was used as additive noise. This was added to the adaptation data at signal-to-noise ratios (SNRs) of 0dB (*BAB00*) and 5dB (*BAB05*).

Convolutional noise was simulated by adding reverberation to the signal using the reverb effect in the digital audio editor, SoX [16]. The data was corrupted with two levels of reverberation: 30% (*RVB30*) and 60% (*RVB60*). The percentages indicate the proportion of output signal occupied by reverberation.

3.1.2. Pre-processing

A way to overcome the problems of noisy adaptation data is to apply signal pre-processing. In a standard custom voice building scenario, a certain amount of background noise is expected in the adaptation data, as well as pops produced by recording with the microphone directly in the airstream. Therefore a pre-processing scheme of silence trimming, high-pass filtering, spectral subtraction and amplitude normalisation was devised.

Pilot experiments showed that with babble and clean data, pre-processing improved the quality of output speech for both CAT and AVM-adapted models. However, with reverberation, there was no significant preference for CAT, and non pre-processed data was preferred for AVM. This could be explained by the fact that the type of pre-processing applied aims to remove additive noise and pops but does not deal with convolutional noise. Therefore, in these experiments, pre-processing was applied for the babble conditions but not for the reverb conditions.

3.2. Parameterisation and label generation

Waveforms were down-sampled to 22050Hz. They were then parameterised using 40 dimensional Mel-LSP coefficients with deltas, log-F0 with first and second order deltas and 20 linear-scale band aperiodicities with deltas. Context feature labels were generated automatically on the clean data.

3.3. Models

3.3.1. AVM model build

The AVM model employs CMLLR and CSMAPLR transforms [1]. A standard training procedure was used: A speaker-independent monophone maximum likelihood model is built and then CMLLR speaker adaptive training is applied. The monophone models are cloned to full context models which are clustered using decision trees. Speaker adaptive training continues with block diagonal global CMLLR transforms for speech, silence and pause. The decision trees, canonical model and global CMLLR transforms are updated several times iteratively. Regression class CMLLR transforms are then trained with the decision trees held fixed and the model parameters updated. The state-duration distributions are treated similarly.

To synthesise a new voice, some samples of the target speaker (adaptation data) are used to create an initial CMLLR transform which is then refined using CSMAPLR followed by a speaker-dependent MAP adaptation of the means so that

$$\hat{\boldsymbol{\mu}}^{(s)}_m = \frac{\tau\boldsymbol{\mu}_m + \sum_{t\in t(s)} \gamma_m(t,s)\left(\hat{\boldsymbol{A}}^{(s)}_{r(m)}\boldsymbol{o}(t) + \hat{\boldsymbol{b}}^{(s)}_{r(m)}\right)}{\tau + \sum_{t\in t(s)} \gamma_m(t,s)} \tag{17}$$

where $\boldsymbol{\mu}_m$ is the mean vector of the AVM; $\gamma_m(t,s)$, $\{\hat{\boldsymbol{A}}^{(s)}_{r(m)}, \hat{\boldsymbol{b}}^{(s)}_{r(m)}\}$ and $t(s)$ are the state occupancy probability, CSMAPLR transforms and data for speaker s respectively, and τ the hyperparameter. The MAP adapted model is combined with the CSMAPLR transform for synthesis.

Potentially, by adding extra freedom, a MAP update of the means may be more susceptible to noisy adaptation data. In initial tests, however, AVM synthesis with and without the final MAP update did not produce noticeably different samples.

3.3.2. CAT model build

A CAT model with six clusters and a bias was built as follows. The AVM canonical model was converted into a speaker-independent model by running one update to remove the CMLLR transforms. This speaker-independent model is copied into the bias cluster. The six additional clusters were initialised with zero means. Each training speaker is assigned to one of the six clusters by perceived similarity[1]. The initial CAT weights were set to 1/0 values corresponding to the speaker's assigned cluster and a value of 1 for the bias. In this way, given (2), the initial CAT model is effectively identical to the speaker-independent model. MDL based decision tree context clustering was performed for each cluster leaving the bias cluster until last. The aim at this stage is to coax the model in such a way that each cluster models speaker specific attributes while the bias cluster models common attributes. Alternative initialisation schemes may be envisaged (e.g. [15]). After context clustering is performed for all CAT clusters, the model's parameters (means and variances) and CAT weights are updated iteratively. Note that the weights are updated independently for each speaker in the training set. The initial grouping of the speakers merely provide a starting point and does not tie the weights of different speakers. The context clustering and iterative model/CAT weight updates are repeated once.

To synthesise a new speaker, an initial set of CAT weights are copied from one of the training speakers. The weights are updated iteratively to maximise the likelihood given the adaptation data. It was observed that the weights converge to the same values irrespective of the starting point.

3.4. Evaluation setup

In order to avoid a walkie-talkie effect resulting from noise being modelled in the start and end silences, CAT weights, CSMAPLR transforms and models for silence were replaced, post-adaptation, with those obtained using clean data. Speech waveforms were synthesised from the generated speech parameters with post-filtering.

Subjective listening tests were conducted via the crowdsourcing website *CrowdFlower* using Mechanical Turk workers located in the US [17]. Listeners were asked to rate the quality of the synthetic speech on a five-point scale where 1 is very bad and 5 is very good. The top end of the scale was anchored with natural speech samples from the same speakers. To anchor the bottom end of the scale, speaker-dependent models were trained for each speaker, with the standard HMM-TTS flat-start approach using the 100 adaptation sentences only.

3.5. Results

Synthesis of the CAT model adapted to noisy data still yielded speech that was clean but with slightly degraded quality. In contrast the AVM's synthesis was corrupted by noise with properties resembling the noise contained in the adaptation data. These observations are consistent with the theory.

Results of the MOS tests are shown in Figure 2. CAT adaptation outperforms the AVM for *CLEAN*, as expected from previous studies [15]. In addition, it is more robust to noise than AVM adaptation; it remains relatively unaffected with increasing levels of noise. There is no significant difference between AVM adapted to clean data and CAT adapted to *BAB05*, *BAB00* or *RVB60*. CAT data adapted to moderate levels of noise can even outperform AVM adapted on clean data, as was seen with CAT *RVB30*.

In contrast, the AVM saw a significant drop in MOS scores with increasing levels of noise, to the extent that AVM *BAB00* is no better than speaker-dependent models trained on the noisy 100 sentences only (SD *BAB00*).

AVM adaptation with noisy data caused the resulting synthesised samples to slow down considerably. For example, the speech of samples synthesised with AVM *BAB00* were on average 51% longer than that synthesised with AVM *CLEAN*. This was due to silent frames for noisy data looking more like speech and thus being consumed by speech models instead of silence or pause models. For example, the proportion of frames assigned to speech (as opposed to silence/pause) during adaptation was 4.8% (relative) higher for *BAB00* than for *CLEAN*.

This resulted in more frames being assigned to speech at synthesis time. As shown in Table 1, different phone types were affected with varying degrees[2]. Voiced obstruents were affected the most (on average 2.6 times longer in *BAB00* than in *CLEAN*). The table shows that silence is the only category for which more frames were allocated to it in the clean condition than in the noisy conditions.

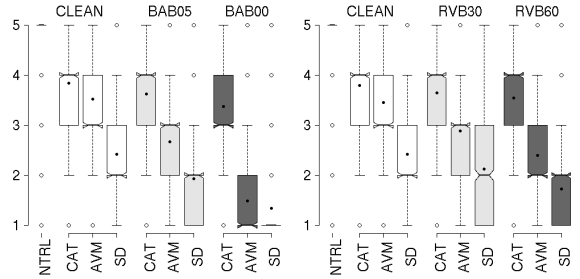
To eliminate any effect on the MOS scores due to the slowing down of the speech, the AVM samples were synthesised using durations obtained from the CAT model and subjective tests re-run. The same overall tendency may be observed in Figure 3: the AVM degrades faster than CAT with increasing noise levels; CAT outperforms the AVM in the presence of noise.

A more constrained form of AVM adaptation with global CMLLR transforms led to less noisy output, compared to AVM with regression class transforms. However, they were still much

phone type	AVM *BAB05*	AVM *BAB00*
voiced obstruents	1.56	2.60
voiceless obstruents	1.21	1.61
nasals	1.37	1.64
other consonants	1.18	1.41
vowels	1.08	1.13
pause	1.26	1.70
silence	0.95	0.95

Table 1: Ratio of average duration per phone for *BAB05*/*CLEAN* and *BAB00*/*CLEAN*, for samples synthesised from the AVM-adapted models, analysed by phone type.

Figure 2: Distribution of mean MOS scores. Black points represent the mean of each distribution.

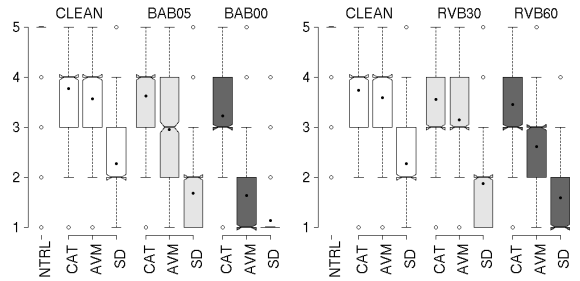


(a) Babble (b) Reverberation

[1]This was based on subjective judgements made by the authors.

[2]Other noises may affect the duration of each phone type differently.

Figure 3: Distribution of mean MOS scores; AVM samples synthesised with CAT duration.



(a) Babble (b) Reverberation

noisier than CAT output. In addition, using global transforms led to more artefacts and decreased similarity.

4. Analysis of speaker similarity in CAT

When building a voice for a target speaker, it is important to assess how similar the synthetic voice is to the original. Under clean conditions, CAT and AVM have been found to perform similarly in terms of speaker similarity [15]. With noisy data for CAT, it was observed informally[1] that speaker similarity degrades with increasing levels of noise, even though the speech quality remained relatively unaffected. The noisier the adaptation data, the more similar the output of different speakers.

It was hypothesised that only a small amount of signal was available for estimating the point in space for the target speaker, due to noise occupying a high proportion of the signal. Thus the speaker subspace was smaller for high levels of noise. This hypothesis may be tested objectively without recourse to subjective tests. In this section, a log-likelihood analysis of the adapted models is performed and multidimensional scaling (MDS) is used to visualise the data.

4.1. Log-likelihood variance analysis

CAT weights for each target speaker were used to align clean data from every other speaker and thus the pairwise log-likelihood of alignment was obtained for each speaker pair. The variance of log-likelihoods was then obtained for each condition as shown in Table 2. It shows that there is less variance in the babble noise conditions than for the clean condition, indicating a smaller speaker subspace. In the reverberation condition, the levels of reverb used in our experiments did not affect the size of the overall subspace as much.

Interestingly, even in noisy conditions, the bimodality of log-likelihoods is retained between male and female speakers. This may be due to log-f0 being relatively unaffected by noise. To test this hypothesis, CAT weights for log-f0 from male speakers were transplanted to female speakers' CAT weights (and vice-versa) and these were used to align data from all the test speakers, in a manner similar to above. This was done for all combinations of speakers.

Analysis was performed by comparing two cases as follows: a) the gender of the data matches the gender of spectral weights but log-f0 weights are of the opposite gender and b) the gender of the data matches the gender of the log-f0 weights but the spectral weights are of the opposite gender. For *CLEAN* the log-likelihood for a) was higher. However, for *BAB00*, b) was marginally higher, confirming our hypothesis that log-f0 plays a big role in determining whether the data aligns better with male or female weights in noisy conditions, as it remains relatively unaffected by noise.

condition	*CLEAN*	*BAB05*	*BAB00*
σ^2 (all)	32.48	25.49	12.21
σ^2 (male)	3.76	3.71	1.81
σ^2 (female)	3.46	2.66	2.09
condition	*CLEAN*	*RVB30*	*RVB60*
σ^2 (all)	27.84	26.67	24.31
σ^2 (male)	4.05	4.19	3.94
σ^2 (female)	3.78	3.85	4.71

Table 2: Variance of each log-likelihood matrix for CAT adaptation. Note that the variances of the *CLEAN* conditions are different because different pre-processing is applied (Section 3.1.2).

4.2. MDS analysis

Another way to investigate the distribution of voices is to visualise them in a low dimensional space derived from the synthesised speech parameters, using an MDS technique [18]. The axes of the space output by MDS do not have any pre-defined meaning, but MDS attempts to preserve the pairwise distances between the speakers, thus placing similar-sounding speakers close to each other in the space.

Parameters were generated from the adapted CAT model, then mean Mel-LSP distances were calculated for each speaker pair. In order to maintain the same number of frames across all speakers, the generated parameters were constrained using the initial duration CAT weights for all speakers. A distance matrix was created for each noise condition and MDS analyses were performed. Figure 4 shows how the speaker space, indicated by the convex hull, shrinks with increasing levels of noise.

5. Conclusion

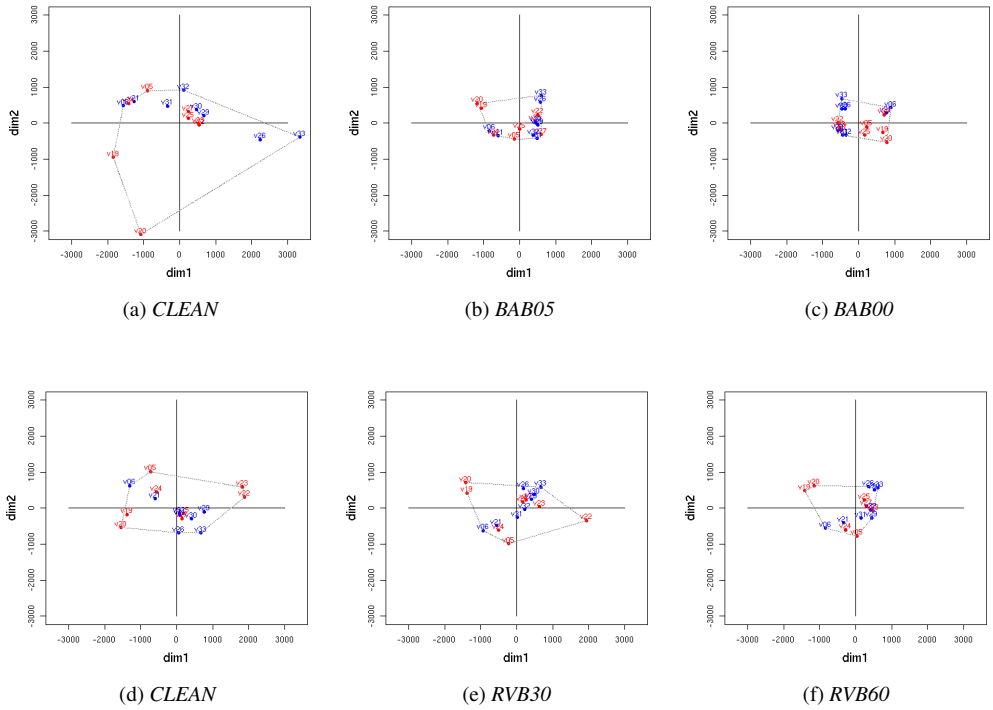
This paper studies the robustness of AVM and CAT adaptation to noisy data. The results of subjective experiments show that AVM suffers significant levels of degradation with noisy adaptation data corrupted with additive noise (babble) and convolutional noise (reverberation). In contrast, in terms of speech quality, CAT is relatively robust to adaptation data with these kinds of corruption.

Pre-processing with spectral subtraction only helps for additive noise and even then there is a limit to how much it can help. While the results would depend on the type and degree of signal processing applied, this indicates that a noise-robust approach to adaptation is still required.

The results confirm the hypothesis that linear transforms used to adapt AVMs are too powerful because noise in the adaptation data is modelled and synthesised in the output speech. The CAT space, on the other hand, is much more constrained so that there is not enough flexibility in the model to deviate much from clean speech, even when the adaptation data is noisy.

This study investigated robustness from the perspective of output speech quality but not speaker similarity. It is known that AVM outperforms CAT in terms of speaker similarity when a large amount of adaptation data is available [15]. With CAT, the speaker subspace became smaller with noisy adaptation data, indicating that speaker similarity is compromised by noise. Subjective evaluation of speaker similarity is left for fu-

Figure 4: MDS visualisation of CAT speaker space computed from Mel-LSP distances between each speaker. Red points represent female speakers and blue points male. See note in Table 2 about pre-processing.



(a) *CLEAN* (b) *BAB05* (c) *BAB00*

(d) *CLEAN* (e) *RVB30* (f) *RVB60*

ture work. Future research will also include the evaluation of a noise-robust approach to produce the context feature labels and to investigate noise factorisation.

6. References

[1] 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," *IEEE Trans. Audio Speech Lang. Process.*, vol. 17, no. 6, pp. 66–83, 2009.

[2] 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," *IEEE Trans. Audio Speech Lang. Process.*, vol. 17, no. 6, pp. 1208–1230, 2009.

[3] J. Yamagishi, M. Lincoln, S. King, J. Dines, M. Gibson, J. Tian, and Y. Guan, "Analysis of unsupervised and noise-robust speaker-adaptive HMM-based speech synthesis systems toward a unified ASR and TTS framework," in *Proc. Blizzard Challenge Workshop*, 2009.

[4] R. Karhila, U. Remes, and M. Kurimo, "HMM-based speech synthesis adaptation using noisy data: analysis and evaluation methods," in *Proc. ICASSP*, 2013.

[5] H. Zen, N. Braunschweiler, S. Buchholz, M. Gales, K. Knill, S. Krstulović, and J. Latorre, "Statistical Parametric Speech Synthesis Based on Speaker and Language Factorization," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 5, pp. 1713–1724.

[6] T. Nose, Y. Kato, and T. Kobayashi, "A speaker adaptation technique for MRHSMM-based style control of synthetic speech," in *Proc. ICASSP*, 2007, pp. 833–836.

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

[8] Y. Ephraim and H. L. V. Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 4, pp. 251–266, 1995.

[9] M. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, 2000.

[10] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov models based on multi-space probability distribution for pitch pattern modeling," in *Proc. ICASSP*, 1999, pp. 229–232.

[11] H. Zen and N. Braunschweiler, "Context-dependent additive log F_0 model for HMM-based speech synthesis," in *Proc. Interspeech*, 2009, pp. 2091–2094.

[12] K. Saino, "A clustering technique for factor analyzed voice models," Master thesis, Nagoya Institute of Technology, 2008.

[13] J. Odell, "The use of context in large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, 1995.

[14] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," in *Proc. Eurospeech*, 1997, pp. 99–102.

[15] V. Wan, J. Latorre, K. K. Chin, L. Chen, M. J. F. Gales, H. Zen, K. Knill, and M. Akamine, "Combining multiple high quality corpora for improving HMM-TTS," in *Proc. Interspeech*, 2012, pp. 1135–1138.

[16] "http://sox.sourceforge.net/."

[17] S. Buchholz, J. Latorre, and K. Yanagisawa, "Crowdsourced assessment of speech synthesis," in *Crowdsourcing for Speech Processing*, M. Eskenazi, G.-A. Levow, H. M. Meng, G. Parent, and D. Suendermann, Eds. Chichester: John Wiley & Sons, 2013, pp. 173–216.

[18] J. Yamagishi, O. Watts, S. King, and B. Usabaev, "Roles of the average voice in speaker-adaptive HMM-based speech synthesis," in *Proc. Interspeech*, 2010, pp. 418–421.