ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Combining multiple high quality corpora for improving HMM-TTS

Vincent Wan, Javier Latorre, KK Chin[1]*, Langzhou Chen,*
Mark J. F. Gales, Heiga Zen[1]*, Kate Knill, Masami Akamine*

Toshiba Research Europe Ltd., Cambridge Research Lab, 208 Science Park, Cambridge, UK
{vincent.wan,javier.latorre,langzhou.chen,mjfg,kate,knill}@crl.toshiba.co.uk
masa.akamine@toshiba.co.jp

Abstract

The most reliable way to build synthetic voices for end-products is to start with high quality recordings from professional voice talents. This paper describes the application of average voice models (AVMs) and a novel application of cluster adaptive training (CAT) to combine a small number of these high quality corpora to make best use of them and improve overall voice quality in hidden Markov model based text-to-speech (HMM-TTS) systems. It is shown that integrated training by both CAT and AVM approaches, yields better sounding voices than speaker dependent modelling. It is also shown that CAT has an advantage over AVMs when adapting to a new speaker. Given a limited amount of adaptation data CAT maintains a much higher voice quality even when adapted to tiny amounts of speech.
Index Terms: speech synthesis, cluster adaptive training, speaker adaptation, average voice models

10.21437/Interspeech.2012-357

1. Introduction

Building high quality speech synthesis voices by employing professional voice talents is an expensive but common strategy. Its benefits are that the sound of the voice, speaking style, recording conditions and other attributes can be controlled and/or specially chosen. Given a set of high quality voice corpora recorded specifically for creating synthetic voices, the standard approach in HMM-TTS is to build speaker dependent models for each speaker separately. This is suboptimal for a number of reasons. First and foremost by exploiting the data available from all speakers it should be possible to create better voices for each speaker. Secondly, with an existing corpus of high quality speakers it should be possible to create a new voice of similar quality with a smaller quantity of data thus reducing the cost of creating new voices.

A simple strategy to combine the data from multiple corpora is by interpolation of the speaker dependent (SD) models [1]. Its advantage is that each voice has its own decision tree and distinct set of parameters. The drawback is that data are not shared during training. Each model is trained on its own portion of the data and commonalities across speakers are not exploited. In essence the data is treated in a fragmented fashion. By itself interpolation does not improve voice quality. It is actually a method of creating new voices but there is no single ideal way of determining the interpolation weights for an existing set of SD models when given a target speaker.

Adaptation based on CMLLR/CSMAPLR average voice model (AVM) speech synthesis [2] approach provides a more efficient way of utilising the training data without fragmenting it. A CMLLR transform maps each speaker into a normalised space in which all speakers may be modelled by a single canonical model.

[1]KK Chin and Heiga Zen are now at Google.

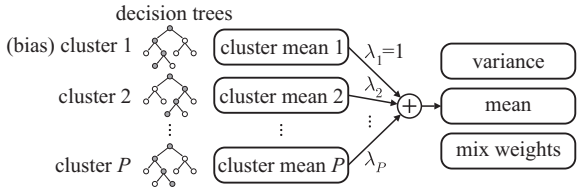

Figure 1: CAT with cluster-dependent decision trees.

Like AVMs, cluster adaptive training (CAT) uses the available data without fragmentation but with the added benefit of having multiple compact decision trees that are interpolated to produce a huge variety of possible contexts which is hard to achieve with a single tree. CAT may be seen as a framework that generalises AVMs. CAT was originally proposed for ASR by [3]. It was extended to HMM-TTS by [4] who used CAT in combination with CMLLR transforms to achieve speaker and language factorisation. There, CAT modelled the language variations while CMLLR transforms modelled the speaker characteristics. Here, CAT alone is used to model speaker characteristics without the aid of CMLLR transforms.

The rest of paper is laid out as follows. Section 2 gives a brief overview of CAT, section 3 describes the experiments and results and section 4 concludes.

2. Cluster adaptive training

The formulation of CAT used in this paper is the same as that described in [4]. The structure of the CAT model is shown in fig. 1. There are multiple clusters each with their own decision tree. The first (bias) cluster is different from the other clusters. The bias cluster contains means, variances and mixture weight parameters whereas the non-bias clusters contain only means. The variances and mixture weights are simply obtained from the bias cluster. To compute the mean μ_m of component m, the decision trees of each cluster are traversed to find the set of cluster specific mean values which are interpolated by a vector of CAT weights $\boldsymbol{\lambda}^{(s)}$ specific to speaker s using the formula

$$\boldsymbol{\mu}_m^{(s)} = \boldsymbol{M}_m \boldsymbol{\lambda}^{(s)} \quad (1)$$

where M is the matrix constructed from the means of the individual clusters and for the bias cluster $\lambda_1 = 1$.

The CAT weights may be interpreted as a vector in an eigenspace representing all possible speakers. Since it is relatively low dimensional, it is easy to move around the space to

produce different sounding voices without recourse to adaptation data. This aspect of controllability in CAT is not present in CMLLR based AVMs. In this way CAT is analogous to speaker interpolation. However, unlike speaker interpolation CAT is not limited to the convex hull defined by the speakers in the training data. We can move to any point in the speaker eigenspace.

In CAT training, the goal is to maximise the log likelihood given the training data and associated transcriptions and speaker labels. An expectation-maximisation (EM) algorithm is used where the canonical parameters (means and variances), the CAT weights and the decision trees are each updated separately in an iterative fashion. The process is similar to that used for speaker adaptive training.

The means of the canonical model are estimated as follows. From the auxiliary function of the EM algorithm the $\boldsymbol{G}_{ij}^{(m)}$ and $\boldsymbol{k}_i^{(m)}$ accumulated statistics are

$$\boldsymbol{G}_{ij}^{(m)} = \sum_{t,s} \gamma_m(t,s) \lambda_{i,q(m)}^{(s)} \boldsymbol{\Sigma}_{v(m)}^{-1} \lambda_{j,q(m)}^{(s)} \tag{2}$$

$$\boldsymbol{k}_i^{(m)} = \sum_{t,s} \gamma_m(t,s) \lambda_{i,q(m)}^{(s)} \boldsymbol{\Sigma}_{v(m)}^{-1} \boldsymbol{o}(t) \tag{3}$$

where $\gamma_m(t,s)$ is the posterior probability of component m generating observation $\boldsymbol{o}(t)$ at time t, $q(m)$ is the regression class of component m and $\boldsymbol{\Sigma}_{v(m)}$ is the covariance matrix for component m determined from the decision tree of the bias cluster. New means are obtained by solving a set of linear equations

$$\begin{bmatrix} \boldsymbol{G}_{11} & \dots & \boldsymbol{G}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{G}_{N1} & \dots & \boldsymbol{G}_{NN} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}}_1 \\ \vdots \\ \hat{\boldsymbol{\mu}}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}_1 \\ \vdots \\ \boldsymbol{k}_N \end{bmatrix} \tag{4}$$

using a sparse matrix solver, where

$$\boldsymbol{G}_{n\nu} = \sum_{\substack{m,i,j \\ c(m,i)=n \\ c(m,j)=\nu}} \boldsymbol{G}_{ij}^{(m)}, \qquad \boldsymbol{k}_n = \sum_{\substack{m,i \\ c(m,i)=n}} \boldsymbol{k}_i^{(m)}. \tag{5}$$

Decision trees are built iteratively on a cluster by cluster basis [5]. When building the trees for one cluster, the other trees and their canonical parameters are held fixed. The tree is built to maximise the log-likelihood given the training data while maintaining a balance between complexity and accuracy. The log-likelihood for the n^{th} node in the i^{th} cluster is given by

$$\mathcal{L}(n) = \frac{1}{2} \hat{\boldsymbol{\mu}}_n^\top \left(\sum_{m \in \mathcal{S}(n)} \boldsymbol{G}_{ii}^{(m)} \right) \hat{\boldsymbol{\mu}}_n. \tag{6}$$

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

$$\hat{\boldsymbol{\mu}}_n = \left(\sum_{m \in \mathcal{S}(n)} \boldsymbol{G}_{ii}^{(m)} \right)^{-1} \times \sum_{m \in \mathcal{S}(n)} \left(\boldsymbol{k}_i^{(m)} - \sum_{j \neq i} \boldsymbol{G}_{ij}^{(m)} \boldsymbol{\mu}_{c(m,j)} \right). \tag{7}$$

So the best question to split the n^{th} node can be selected based on the log-likelihood gain.

The CAT weights are estimated using the methods described in [3] and [6].

3. Experiments

3.1. Data

The speech data are recorded from professional voice talents in studio conditions. Each speaker spoke in the General American accent of US English with a neutral style. Two male and two female speakers were used for training. Table 1 shows the amount of speech data (i.e. excluding silence) available for each speaker. A third female speaker (FAK) with 1.5 hours of speech data was available for adaptation experiments. Two randomly chosen subsets of FAK were also defined, each having 6 seconds and 10 minutes of speech. The waveforms, sampled at 16kHz, were parameterised using 40 dimensional Mel-cepstral coefficients, log-F0 and 21 Bark-scale band aperiodicities, each with their first and second order deltas.

3.2. Experimental setup

All samples were synthesised by a speech generation algorithm including a global variance (GV) term [7]. Where the synthesis was for a voice within the training data, a speaker specific GV was used otherwise a speaker independent GV, estimated from the four training speakers, was used instead. Speech waveforms were synthesised from the generated speech parameters using the source-filter model.

Subjective listening tests were conducted via the crowdsourcing website *CrowdFlower* using Mechanical Turk workers located in the US [8]. Standard paired-comparison preference tests asked listeners to choose the waveform that sounded better. Listeners were given the option of choosing "no preference".

In the ABX similarity listening tests, subjects were first played a reference sample from the voice talent followed by two synthesised samples. They were asked to choose the sample that sounded more similar to the speaker in the reference. In this case, listeners could not choose "undecided".

Mean opinion score (MOS) tests ask listeners to rate the quality of the synthetic speech on a five point scale where 1 is very bad and 5 is very good.

3.3. Baselines

Two baseline systems are trained. The first baseline is the set of four speaker dependent (SD) models each trained on one of the training speakers. The SD models are trained using the standard HMM-TTS flat-start approach.

The second baseline is an AVM employing CMLLR and CSMAPLR transforms [2]. A single AVM is trained on the four training speakers thus: a speaker independent monophone maximum likelihood model is built 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 in the same way. To synthesise the training speakers, the CMLLR transforms estimated during training are refined using CSMAPLR. This is followed by a speaker dependent MAP adaptation of the means. The MAP adapted model is combined with the CSMAPLR transform for synthesis. To synthesise a new speaker, some samples of the voice must be provided to create an initial CMLLR transform which is then refined using the same CSMAPLR/MAP adaptation process as before.

3.4. CAT model build

A CAT model with 4 clusters and a bias was built. The FLH speaker dependent model was used to initialise the bias cluster. Four additional clusters were used to represent each speaker as listed in the order shown in table 1. The initial CAT weights were set to one/zero values corresponding to the speaker's cluster assignment. The bias cluster has a fixed weight of 1. Decision tree context clustering (MDL based, 10-fold cross-validation) was performed for each cluster in the following order: MGT, FSP, MMJ, bias then FLH. The statistics required for context clustering were obtained from the respective speaker dependent model. For the bias cluster this meant choosing the speaker dependent model according to the source of the utterance. Updating the clusters in this way is an attempt to force the clusters to model speaker specific attributes and the bias to model the common attributes[1]. After context clustering is performed for all CAT clusters then the model's parameters (means and variances) and CAT weights are iteratively updated. Subsequent rebuilding of the decision tree context clustering was done in order from left to right in table 1 without use of the speaker dependent models. The context clustering and iterative model/CAT weight updates were repeated twice more.

The final set of CAT weights estimated during training were used to synthesise samples for the training speakers. To synthesise a new speaker an initial set of CAT weights were copied from one of the training speakers. They were then repeatedly updated to maximise the likelihood given the adaptation data until convergence. The CAT weights converged to the same values irrespective of the starting point.

Examining the number of leaf nodes in the decision trees of the CAT model's clusters can give insights into the data. The relative values indicate how information is shared across the different clusters. Table 1 shows the size of each cluster's trees for each stream. Roughly, there is a correlation between the amount of data for each speaker and the size of the decision trees on which those speakers were initialised.

Shared information tends to be captured by the bias cluster while other clusters capture more speaker specific information. Thus the ratio of the number of bias to non-bias leaf nodes indicates the amount speaker specific information that exists in each stream. In the MGC stream that ratio is much larger here than those reported in [4]. In that work a CMLLR transform was used to model speaker variety and CAT for language variety. Here the CAT clusters must model speaker variety directly thus the non-bias cluster trees are larger here. The ratio for the LF0 trees is the greatest suggesting that the prosody is much more speaker dependent. In contrast, the bias cluster trees of MGC and BAP are larger than the non-bias clusters suggesting that there is a more information shared across speakers which can be captured by the bias cluster.

3.5. Evaluating training speakers

To compare the voices obtained for the training speakers, 100 test sentences were synthesised from each of the SD, AVM and CAT models and compared using paired-comparison preference tests. The overall results are shown in table 2. There is a clear preference for CAT and AVM over SD modelling. This clearly shows that combining multiple corpora into a single integrated training produces better sounding synthetic voices.

Comparing AVM with CAT, there is a small but overall not significant preference towards the CAT model. More detailed results split by speaker are in table 3. The CAT model is significantly preferred for two out of the four speakers, the AVM model is preferred for one out of the four and no preference in the last case. Fig. 2 shows the results of a MOS test comparing SD, AVM and CAT approaches for each of the training speakers. Both AVM and CAT have similar MOS and both are up to 0.7 MOS points higher than the SD models for some speakers.

[1]The CAT model could also have been initialised from a speaker independent model.

	bias	MGT	FSP	MMJ	FLH
Hours of speech		2.0	3.5	1.0	4.2
MGC	3885	1550	1730	934	2531
LF0	7859	21142	23288	19974	20521
BAP	8422	1174	1211	769	1480

Table 1: Amount of training data and number of leaf nodes in the CAT model's trees.

SD	AVM	CAT	No pref	p
32.3	54.0	-	13.8	< 0.001
30.5	-	56.1	13.4	< 0.001
-	40.2	45.1	14.7	0.076

Table 2: Overall preference test results for training speakers.

3.6. Building a new voice

To create a new "high quality" voice, the standard method is to record a new corpus and train a speaker dependent model using a flat-start approach. There are two approaches available for AVM and CAT models. The first is to fold the new data in with the existing data and retrain the model. This is a similar scenario to that in section 3.5 and it is expected that adding new data in this way will also improve the existing voices. The other approach is to adapt the model to the new data. This is the approach investigated here. The AVM and CAT models were adapted using the complete 1.5 hours of FAK adaptation data. Since collecting high quality data is expensive, adapting on the 10 minute and 6 second subsets is investigated also. The baseline SD model here is always trained on the complete 1.5 hours of data. No SD models were trained on the FAK subsets. Preference and ABX listening tests were performed and results are shown in tables 4 and 5 respectively. The tests show consistently that, for this speaker, the CAT model is preferred over SD and AVM when adapting to 1.5 hours of data. In terms of similarity to reference speaker, the ABX results on 1.5 hours indicate that the CAT model produces the most similar sounding voice. When the amount of adaptation data is reduced, the voice similarity drops quickly for the AVM whereas the CAT model is much more robust even for the 6 seconds case. This result may be due to the number of parameters that need to be estimated during adaptation. Many more parameters need to be estimated for the AVM than for the CAT model. Results of a MOS test comparing the AVM and CAT adapted to various amounts of data with the speaker dependent model trained in 1.5 hours is shown in fig. 3. On this data CAT achieves a higher MOS than both SD trained on 1.5 hours and AVM. Moreover, the CAT model's MOS remains high irrespective of the amount of adaptation data.

Speaker	AVM	CAT	No pref	p
MGT	**58.6**	32.6	8.8	< 0.001
FSP	31.1	**53.8**	15.2	0.005
MMJ	34.5	**57.6**	7.9	0.001
FLH	34.9	42.6	22.5	0.189

Table 3: Preference test results for training speakers: AVM vs. CAT by speaker.

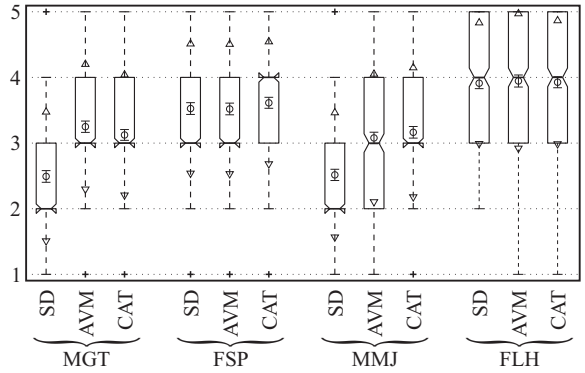

Figure 2: MOS test results for the training speakers

4. Conclusion

This paper compares the application of speaker adaptive training methods to the standard speaker dependent modelling approach for building high quality HMM-TTS voices using speech recorded by professional voice talents in well equipped recording studios.

AVM and CAT approaches were evaluated and although it was found that both approaches were able to produce better synthesis than speaker dependent models, the CAT approach yielded better voice quality and was preferred by listeners for most voices. When creating a new voice by adaptation, it was shown that CAT can achieve a better voice than both AVM and speaker dependent modelling approaches while requiring significantly less speech data. We acknowledge that the adaptation results are currently based upon a single speaker. Running additional adaptation tests on other high quality corpora will require additional data collection.

In addition to adapting the CAT weights to new speakers, it is possible to adapt a CAT model by adding an additional cluster with its own context-dependent decision tree specifically estimated to maximise the log-likelihood given the adaptation data. Alternatively, by further applying a CMLLR transform during training the CAT model becomes a generalisation of an AVM with the advantage of having multiple trees in the canonical model. These shall be the focus of future work. Further extensions of CAT can also be found in [9] and [10].

SD	AVM	CAT	No pref	p
47.8	50.2	-	1.9	0.365
30.8	-	68.3	1.0	< 0.001
-	28.4	67.2	4.5	< 0.001

Table 4: Preference test. Models trained or adapted on 1.5 hours of FAK data.

Adapt data	SD	AVM	CAT	p
1.5 hours	56.7	43.3	-	0.004
	45.0	-	55.0	0.016
	-	46.0	54.0	0.061
10 mins	73.8	26.2	-	< 0.001
	44.4	-	55.6	0.005
	-	27.3	72.7	< 0.001
6 secs	94.7	5.3	-	< 0.001
	56.8	-	43.2	0.001
	-	5.9	94.1	< 0.001

Table 5: ABX similarity to reference after adapting to varying amounts FAK data. The reference SD model was trained on 1.5 hours for all cases.

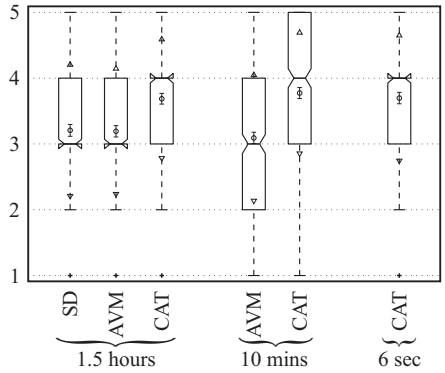

Figure 3: MOS test results FAK adaptation.

5. References

[1] M. Orhan and C. Demiroglu, "HMM-based text to speech system with speaker interpolation," in *2011 IEEE 19th Conference on Signal Processing and Communications Applications (SIU)*, April 2011, pp. 781–784.

[2] 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, pp. 66–83, 2009.

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

[4] 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, 2012.

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

[6] K. Sim and M. Gales, "Adaptation of precision matrix models on large vocabulary continuous speech recognition," in *Proc. ICASSP*, 2005, pp. 97–100.

[7] T. Toda, A. Black, and K. Tokuda, "Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter," in *Proc. ICASSP*, 2005, pp. 9–12.

[8] S. Buchholz and J. Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *Proc. Interspeech*, 2011.

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

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