Asynchronous Factorisation of Speaker and Background with Feature Transforms in Speech Recognition

Oscar Saz, Thomas Hain

Speech and Hearing Group, University of Sheffield, Sheffield UK

O.SazTorralba@sheffield.ac.uk, T.Hain@dcs.shef.ac.uk

Abstract

This paper presents a novel approach to separate the effects of speaker and background conditions by application of feature–transform based adaptation for Automatic Speech Recognition (ASR). So far factorisation has been shown to yield improvements in the case of utterance-synchronous environments. In this paper we show successful separation of conditions asynchronous with speech, such as background music. Our work takes account of the asynchronous nature of the background, by estimation of condition-specific Constrained Maximum Likelihood Linear Regression (CMLLR) transforms. In addition, speaker adaptation is performed, allowing to factorise speaker and background effects. Equally, background transforms are used asynchronously in the decoding process, using a modified Hidden Markov Model (HMM) topology which applies the optimal transform for each frame. Experimental results are presented on the WSJCAM0 corpus of British English speech, modified to contain controlled sections of background music. This addition of music degrades the baseline Word Error Rate (WER) from 10.1% to 26.4%. While synchronous factorisation with CMLLR transforms provides 28% relative improvement in WER over the baseline, our asynchronous approach increases this reduction to 33%.

Index Terms: Speech recognition, adaptation, factorisation, asynchronous decoding

1. Introduction

State–of–the–art Automatic Speech Recognition (ASR) algorithms often lack robustness in natural situations. Different and varying acoustic environments are one of the main factors in the degradation of the performance of ASR systems. Systems that perform properly in controlled conditions may become not suitable in the presence of other sound sources. Research has focused mostly on situations where the background is synchronous with the utterance, generally background noise. To treat interfering signals as generic noise has its disadvantages, and hence more general adaptation techniques are used in these environments. Solid results have been achieved with techniques operating in the model space, such as Parallel Model Combination (PMC) [1] or Maximum Likelihood Linear Regression (MLLR) [2], or in the feature space, such as Constrained MLLR (CMLLR) [3], Stereo-based Piecewise Linear Compensation for Environment (SPLICE) [4] or Multi-Environment Model-based LInear Normalization (MEMLIN) [5].

In recent works, factorisation was used as a way to further improve the performance of ASR in mixed conditions [6]. Techniques based on joint factorisation of sources of variability, for example Joint Factor Analysis (JFA) [7] as used in speaker verification tasks, are now being considered for use in ASR. Subspace Gaussian Mixture Models (SGMMs) incorporate this joint factorisation idea [8]. Other approaches are based on combining transforms for the speakers and the environments in a joint way. This has been done with Vector Taylor Series (VTS) and MLLR transforms in [9], CMLLR transforms in [10] and CMLLR and MLLR transforms in [11]. The joint training of speaker and background transforms has been shown to be more robust to changes in the background conditions.

However, for all methods so far it has been assumed that the environment for any input signal is maintained throughout the utterance. That assumption is true for corpora like Aurora [12], where one single type of noise was added to each speech signal. In a more natural situation however this assumption is often not valid. In media data, beyond the traditional and very controlled broadcast news scenario, one can expect a number of events which are completely independent and hence asynchronous with the spoken words. These events can be background music in music shows, special effect sounds in drama shows, applause in live shows or quiz shows. In other tasks as meeting transcription asynchronous events like laughter, door slamming, etc may be present. The common feature of such events is that their occurrence is not tied to the beginning and end of a speech utterance, hence modelling them as a single static environment will be suboptimal.

Work presented in this paper follows up the idea of factorising speaker and background, but generalising to asynchronous conditions. Environment transforms will be learned from different sections of the input speech signal, and these will be used to jointly learn a set of speaker transforms. Finally, in the decoding stage, the transforms will be applied also asynchronously to compensate for the changing background.

The paper is organised as follows: Section 2 presents a review on adaptation and factorisation methods with CMLLR transforms. Section 3 outlines our proposed method for asynchronous factorisation and asynchronous decoding. Section 4 describes the experimental setup using WSJCAM0, with results discussed in Section 5. Finally, Section 6 provides discussion and conclusions to this work.

2. Adaptation and factorisation

CMLLR is an adaptation technique typically used for adapting Hidden Markov Models (HMMs) to a specific speaker. By employing a linear transform to both mean and covariance CMLLR can be equally interpreted as a feature transform, which is very useful in many practical situations. A transformation matrix (A) and a bias vector (b) are estimated from data from the desired speaker. Afterwards, given an input feature vector x it transforms it to the vector y which is used in decoding.

$$y = Ax + b \tag{1}$$

The CMLLR transform for a speaker spk is defined as the pair of transformation matrix and bias vector $W_{spk} = \{A_{spk}, b_{spk}\}$ from Equation 1. The transform can also be trained on all the utterances from different speakers in a given environment env, providing an environment transform: $W_{env} = \{A_{env}, b_{env}\}$. CMLLR can be used in supervised adaptation, with manually transcribed data, or in unsupervised adaptation, using the output of a a first pass recognition stage.

2.1. Factorisation with CMLLR transforms

A method for factorising environment and speaker variability was proposed in [10], by means of CMLLR transforms trained in cascade. The method proposed to train an environment transform $W_{env} = \{A_{env}, b_{env}\}$ for every possible environment and across all speakers. These transforms are then used as parent transforms when training speaker transforms $W_{spk} = \{A_{spk}, b_{spk}\}$ for each speaker and across all environments. Thus, given a utterance signal x, spoken by speaker spk in environment env, the observations are transformed to y:

$$y = A_{spk}(A_{env}x + b_{env}) + b_{spk} \quad (2)$$

Using both transforms in cascaded fashion was shown to improve results over conventional CMLLR adaptation on environment and speaker. Also, the speaker transforms, which had been decoupled from the environment, were shown to perform well when used across different environments.

3. Asynchronous factorisation of speaker and environment

One of the issues with factorisation as outlined above is the applicability of transforms to the complete utterance, as conditions often change within a segment of speech. Describing this situation with a single transform is clearly suboptimal and will lead to imprecise modelling. Here we propose that better results can be obtained by identifying these environments asynchronously within the speech, and then learn different environment transforms. Once appropriate environment transforms are found, standard speaker adaptation can be performed in conjunction.

Hence each frame requires classification as belonging to one of the possible environments. Different CMLLR environment transforms $W_{env} = \{A_{env}, b_{env}\}$ are then generated from each group of frames across all speakers. Then, these transforms are applied to the frames of the input signal according to their acoustic environment classification and a set of speaker CMLLR transforms are then trained.

In the decoding stage, the input feature vector is transformed according to Equation 3; for each frame t the most likely environment transform $W_{env}(t) = \{A_{env}(t), b_{env}(t)\}$ is applied, and then the corresponding speaker transform $W_{spk} = \{A_{spk}, b_{spk}\}$ is applied on top.

$$y(t) = A_{spk}(A_{env}(t)x(t) + b_{env}(t)) + b_{spk} \quad (3)$$

3.1. Asynchronous decoding with CMLLR transforms

A key issue in this approach is the way in which the presence of a specific condition in a frame is determined. A classifier can be built separately trained on supervised data. However, in practice automatic on–line classification will often be required. One option is to include such decisions in the decoding process itself. The advantage of such an approach is consistency with the actual recognition, however it can potentially lead to some undesired effects. In this approach, all environment transforms can be used during the decoding stage, and is the decoder itself who can select the most likely label for each frame, by aiming to maximise the overall likelihood.

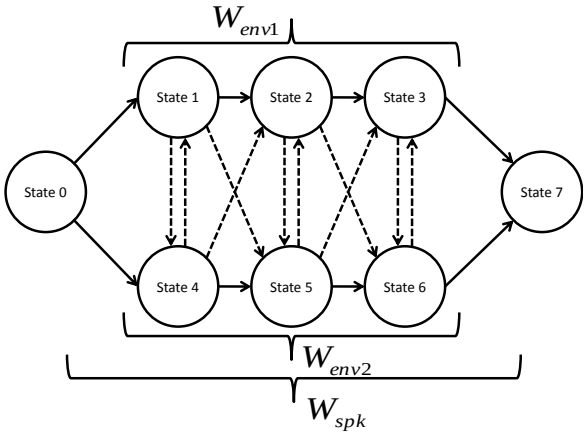

Figure 1: *Asynchronous topology with two environment transforms (W_{env1} and W_{env2}) and a speaker transform (W_{spk}) (auto transitions have been removed for clarity).*

Figure 1 presents how the asynchronous decoding works when there are two possible environment transforms, W_{env1} and W_{env2}, with a speaker transformation W_{spk} on top. This approach naturally generalises to any limited number of environment transforms. The usual three–state left–to–right topology with states 0 and 4 as entry and exit states respectively is modified to include 3 extra states. Now state 7 is the exit state and states 4, 5 and 6 are replicas of the states 1, 2 and 3 respectively. However, states 1, 2 and 3 are associated to the environment transform W_{env1} and states 4, 5 and 6 to the transform W_{env2}. The same speaker transform is used for all the states.

The topology in Figure 1 is *Fully asynchronous*, since it allows transitions among the two possible transforms from one frame to the next. This topology could be restricted to be *Phone–synchronous* by removing all the dashed transitions. In this case, the environment transform can not be switched inside a phoneme, but can be switched in the transition to another phoneme. This choice of topologies will be studied in our experimental work.

This frame–by–frame asynchronous decoding is similar to the proposal for on-line Vocal Tract Length Normalization (VTLN) in [13]. In that work, the model space was augmented to consider different VTLN warping values, and the decoder automatically chose the path that maximized the total likelihood through the augmented space.

3.2. Asynchronous training of CMLLR transforms

The same topology presented in Figure 1 can be used when learning transforms from adaptation data. This topology aligns the input speech to the best sequence of states which can change asynchronously among environments. Afterwards, a new set of transforms can be updated from the alignment statistics.

However, it is required to already have an initial set of environment transforms. In our work, which will be based on two possible environments, we will initialise one of the environment transforms to be the identity ($W = \{I, 0\}$) and the other to be a single-class CMLLR transform trained on all the adaptation data. These initial transforms can be, then, reestimated to model both the clean and corrupted parts of the input signals respectively. Once the environment transforms are calculated, it is direct to apply this topology to jointly train speaker transforms.

4. Experimental setup

In this paper, we will evaluate the proposed methods with a modified version of the WSJCAM0 corpus. WSJCAM0 was recorded by CUED in 1994 to provide a resource similar to the original WSJ corpus, but with a British English pronunciation [14]. Equivalently, it defines sets for training, development and evaluation. In our work, we used the original speaker independent training set (si_tr) of 7,861 utterances for model training and the 4 development sets (si_dt5a, si_dt5b, si_dt20a and si_dt20b), with 368, 374, 361 and 368 utterances each, for testing. The first two test sets are designed as a 5,000-word closed vocabulary task with a bigram language model, the reamining two sets are designed for a 20,000-word open vocabulary task with a trigram language model. The test sets contain the same 20 speakers, and can be used for unsupervised adaptation.

We created equivalent corrupted test sets, where bursts of music were added in the following manner: A group of 25 pieces of instrumental orchestral music was taken as source for the background music. Each speech signal in the original test sets was contaminated with a burst of music randomly chosen from the source music pieces. These bursts had a uniform random length between 0 and the total length of the clean signal. Any music segment was scaled randomly, but to ensure that its overall energy was between 5 and 15dB below the energy of the overlapping speech signal (to avoid dominance of the music signal). Also, a simple fade effect was used at the very beginning and end of the music burst, to avoid signal discontinuities. In these corrupted test sets, 48.8% of the total frames will contain some level of music in the background. From now on, we will call these different test sets as *Clean* for the original set, and *Music* for the set contaminated with music.

The ASR system was based on a Hidden Markov Model Toolkit (HTK) [15] setup. Crossword triphone models were trained using Maximum Likelihood (ML) from the training set, with 16 Gaussian mixtures per state. We used 39-dimension feature vectors with 13 Perceptual Linear Predictive (PLP) features [16] and their first and second derivatives. Cepstral Mean Normalization (CMN) was applied to the static features.

4.1. Baseline results

The baseline word error rates (WER) of our system on the WSJCAM0 set are shown on Table 1. The results on *Clean* data show an average WER of 6.3% on the 5K tasks, and 13.9% on the 20k tasks. Using the *Clean* models on the sets corrupted by *Music*, the results are significantly worse, with an increase of 16.3% in WER. To provide a baseline with properly matched models, additional acoustic models were trained. The training set was modified to include background music in similar ways to the test sets. A different set of 67 music pieces was used to ensure that the music patterns in the training set would not reappear in the test sets. Table 1 shows that, without adaptation, the best performance is obtained with models trained on *Music* data. While the global WER is 6.5% poorer than for the *Clean* case, it is 9.8% lower than the mismatched case.

Table 1: *Baseline WER on the WSJCAM0 corpus, with models trained and tested in the* Clean *and* Music *conditions.*

Train	Test	5K sets	20K sets	Total
Clean	Clean	6.3%	13.9%	10.1%
Clean	Music	21.0%	31.8%	26.4%
Music	Music	11.3%	21.7%	16.6%

5. Results

This section presents the results of the use of CMLLR transforms, factored CMLLR transforms and asynchronous factored CMLLR transforms. The models were trained on *Clean* data and adaptation was performed in unsupervised fashion.

5.1. CMLLR adaptation

The results when using CMLLR speaker transforms are shown in Table 2. All experiments are based on a regression class tree with 4 classes. Speaker adaptation reduces the error rate by 10% relative for *Clean* data and 30% for the *Music* data. The second part of Table 2 presents results when applying these speaker transforms in mismatched conditions, i.e. transforms derived from *Clean* or *Music* data, applied to *Music* and *Clean* data respectively. In both cases there is some benefit (compared to results in Table 1), but matched adaptation gives significantly better results. It is this effect what first prompted work on factorised transformations.

Table 2: *WER with CMLLR speaker transforms trained on the* Music *data and tested on* Clean *and* Music *data.*

Test	Transform	5K sets	20K sets	Total
Matched conditions				
Clean	Clean	5.4%	12.7%	9.1%
Music	Music	13.6%	23.2%	18.5%
Mismatched conditions				
Clean	Music	5.6%	13.2%	9.5%
Music	Clean	19.7%	30.4%	25.1%

5.2. Factored CMLLR adaptation

We applied the recipe in [10] to our setup in the following way. We trained an environment transform from all the utterances in the *Music* test sets, and used it as a parent transform when training the speaker transforms. We used a regression tree of 2 classes for the environment transform and 2 classes for the speaker transforms, which gives the same number of parameters to learn, making results comparable with CMLLR adaptation.

The results of the evaluation of the different transforms in the *Clean* and *Music* data are shown in Table 3. They show that the joint use of both transforms does not improve the results of having a single speaker transform on the *Music* set; this can be explained by the fact that having defined a single environment, the two transforms are effectively working as one. When using the speaker transforms alone, which have had the environment influence factored out, we see solid improvement over the baseline. In the case of using these factored speaker transforms trained from the *Music* set in the *Clean* set it reaches the improvement achieved by the CMLLR transforms trained on *Clean* data. As it was shown in [10] and [11], speaker transforms trained in a factored approach achieve good results in mismatched conditions.

Table 3: *WER with factored CMLLR transforms trained on the* Music *data and tested on* Clean *and* Music *data.*

Test	5K sets	20K sets	Total
Speaker and environment transforms			
Music	13.6%	23.8%	18.8%
Speaker transforms			
Clean	5.4%	12.6%	9.0%
Music	17.8%	28.2%	23.1%

Table 4: *WER with asynchronous factored CMLLR speaker and background transforms from* Music *data used on* Clean*,* Music *data.*

Adaptation	Decoding	Test	5K sets	20K sets	Total
Phone synchronous	Phone synchronous	Clean	5.9%	12.8%	9.4%
		Music	13.3%	22.8%	18.1%
Phone synchronous	**Fully asynchronous**	Clean	5.8%	12.6%	**9.2%**
		Music	12.6%	22.8%	**17.7%**
Fully asynchronous	Phone synchronous	Clean	5.8%	12.7%	9.3%
		Music	13.4%	23.1%	18.3%
Fully asynchronous	Fully asynchronous	Clean	5.9%	12.6%	9.3%
		Music	12.9%	23.0%	18.0%

5.3. Asynchronous factored CMLLR adaptation

A set of experiments was conducted to investigate our proposed method for asynchronous factorisation of speaker and backgrounds (Section 3). To keep results comparable, we used a single regression class for each of the two environment transforms and two classes for the speakers. Hence the same number of parameters are used when compared to previous situations.

First the effect of the two model topology options is studied. Both the *Phone–synchronous* and the *Fully asynchronous* topologies can be used in either adaptation or decoding, thus leading to four possible configurations The results with speaker and environment transforms are shown in Table 4 for these four cases. The best result is achieved with the *Phone–synchronous* topology in adaptation and *Fully asynchronous* topology in decoding. In the *Music* data, WER reductions of 0.8% compared to CMLLR and 1.1% compared to factored CMLLR are obtained, which represents a 5% in relative WER reduction. In the *Clean* set, the use of both transforms yields a WER of 9.2%, which shows that the asynchronous decoding works successfully even in the presence of only one background condition.

Table 5 presents results when using the speaker transforms learned in the asynchronous setup, without application of the environment transforms. This allows to study the influence of speaker adaptation only on the final results. The best results are obtained with the *Phone–synchronous* model topology during adaptation, yielding 9.0% and 19.5% on the *Clean* and *Music* data respectively. This indicates good factorisation of the speaker, as the WER improvement is only 4% lower than when applying CMLLR adaptation.

Table 5: *WER with asynchronous factored CMLLR speaker transforms from* Music *data used on* Clean*,* Music *data.*

Adaptation	Test	5K sets	20K sets	Total
Phone synchronous	Clean	5.5%	12.5%	**9.0%**
	Music	14.4%	24.5%	**19.5%**
Fully asynchronous	Clean	5.5%	12.6%	9.1%
	Music	14.7%	25.0%	19.9%

5.4. Frame classification with the asynchronous topology

To understand how the proposed model topologies are separating speech from speech with background music, one can investigate the effective frame classification performance. We studied the output state sequence provided by the asynchronous topology on the test sets, as the state sequence implicitly holds information on the environment selected for each frame. We can then compare this distribution of environments with ground truth information which is very precise due to the approach taking in defining the test conditions.

Table 6 shows the results of such comparisons, in the form of the percentage of frames correctly assigned to the class of being speech or speech with music. The frame accuracy for *Phone–synchronous* topology and *Fully asynchronous* topology are 82% and 76% respectively. Typically classifiers for acoustic events (see e.g. [17]) require constraints on transitions. However despite frame by frame decisions the asynchronous topology is able to keep track of the correct speech background. This allows for the improvements in recognition detailed above.

Table 6: *Accuracy in framewise classification.*

	Phone-synchronous	Fully-asynchronous
Speech	83.6%	76.2%
Speech & music	80.5%	76.9%
Global accuracy	82.1%	76.5%

Table 6 can also explain why the best combination of topologies is *Phone–synchronous* in adaptation and *Fully asynchronous* in decoding. The *Phone–synchronous* topology provides a smoother transition between environments and is better in deciding which is the background condition of a frame. That seems to benefit in the training of the environment transforms. In decoding, the *Fully asynchronous* topology makes more frame errors, but provides a higher degree of freedom to the decoder, whose goal is not to maximise the classificaton accuracy but to maximise the overall likelihood.

6. Conclusions

In this work we have proposed a new method for asynchronous factorisation with CMLLR transforms and have shown that it is helpful in adaptation to dealing with speech corrupted by asynchronous bursts of background music. The results presented show a additional significant WER reduction, from 18.5% to 17.7%, in comparison with a standard synchronous approach. The better description of the acoustic environment also helps in more effective factorisation of the speaker variability. Speaker transforms now become usable across different, even unseen, environments.

The paper further investigated a method for asynchronous application of transforms in decoding. The proposed framework can be used for switching of transforms in an asynchronous manner to input speech. The results, presented here for two transforms, will easily generalise to more transforms by an increase in the number of branches in topology proposed.

In the future, this framework can be applied in more realistic data where asynchronous acoustic events occur naturally. In media data, for instance, background music is a common feature and the application of asynchronous factorisation can be expected to improve recognition performance.

7. Acknowledgements

This work was supported by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

8. References

[1] Gales, M. J. F., and Young, S. J., "Robust continuous speech recognition using parallel model combination", *IEEE Trans. on Speech and Audio Processing, 4(5)*, pp. 352–359, 1996.

[2] Gales, M. J. F., and Woodland, P. C., "Mean and Variance Adaptation within the MLLR Framework", *Computer, Speech and Language, 10(4)*, pp. 249–264, October 1996.

[3] Gales, M. J. F., "Maximum Likelihood Linear transformations for HMM-based speech recognition", *Computer, Speech and Language, 12*, pp. 75–98, 1998.

[4] Droppo, J., Deng, L. and Acero, A., "Evaluation of the SPLICE Algorihtm on the AURORA2 Database", in *Proc. of Eurospeech 2001*, 217–220, Aalborg, Denmark.

[5] Buera, L., Lleida, E., Miguel, A., Ortega, A. and Saz, O., "Cepstral Vector Normalization based on Stereo Data for Robust Speech Recognition", *IEEE Trans. on Audio, Speech and Language Processing, 15(3)*, pp. 1098–1113, 2007.

[6] Gales, M. J. F., "Acoustic factorisation", in *Proc. of ASRU 2001*, Madonna di Campiglio, Italy.

[7] Yin, S. C., Rose, R. C. and Kenny, P., "A Joint Factor Analysis Approach to Progressive Model Adaptation in Text-Independent Speaker Verification", *IEEE Trans. on Audio, Speech and Language Processing, 15(7)*, pp. 1999–2010, 2007.

[8] Povey, D., Burget, L., Agarwal, M., Akyazi, P., Kai, F., Ghoshal, A., Glemberk, O., Goel, N., Karafiat, M., Rastrow, A., Rose, R. C., Schwarz, P. and Thomas, S., " The Subspace Gaussian Mixture Model - A Structured Model for Speech Recognition", *Computer, Speech and Language, 25(2)*, pp. 404–439, April 2011.

[9] Wang, Y., and Gales, M. J. F., "Speaker and Noise Factorisation on the AURORA4 Task", in *Proc. of ICASSP 2011*, Prague, Czech Republic.

[10] Seltzer, M. L. and Acero, A., "Separating Speaker and Environmental Variability Using Factored Transforms", in *Proc. of Interspeech 2011*, pp. 1097–1100, Florence, Italy.

[11] Seltzer, M. L. and Acero, A., "Factored Adaptation using a Combination of Feature-space and Model-space Transforms", in *Proc. of Interspeech 2012*, Portland OR, USA.

[12] Hirsch, H. G., and Pearce, P., "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions", in *Proc. of ISCA ITRW ASR*, Paris, France, 2000.

[13] Miguel, A., Lleida, E., Rose., R., Buera, L., Saz, O. and Ortega A.: "Capturing Local Variability for Speaker Normalization in Speech Recognition", *IEEE Trans. on Audio, Speech and Language Processing, 16(3)*, pp. 578–593, 2008.

[14] Robinson, T., Fransen, J., Pye, D., Foote, J. and Renals, S., "WSJCAM0: A British English Speech Corpus for Large Vocabulary Cotniuous Speech Recognition", in *Proc. of ICASSP 1995*, pp. 81–84, Detroit MI, USA.

[15] Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J. J., Ollason, D. G., Povey, D., Valtchev, V., and Woodland, P. C., "The HTK Book version 3.4", Cambridge University Engineering Department, Cambridge, UK, 2006.

[16] Hermansky, H., "Perceptual linear predictive (PLP) analysis of speech ", *J. Acoust. Soc. Am. 87(4)*, pp. 1738–1752, 1990.

[17] Castan, D., Ortega, A., Villalba, J., Miguel, A. and Lleida, E., "Segmentation-by-classification system based on Factor Analysis", in *Proc. of ICASSP 2013*, Vancouver, Canada.