Discriminatively Trained Features Using fMPE for Multi-Stream Audio-Visual Speech Recognition

Jing Huang and Daniel Povey

IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{jghg,dpovey}@us.ibm.com

Abstract

fMPE is a recently introduced discriminative training technique that uses the Minimum Phone Error (MPE) discriminative criterion to train a feature-level transformation. In this paper we investigate fMPE trained audio/visual features for multi-stream HMM-based audio-visual speech recognition. A flexible, layer-based implementation of fMPE allows us to combine the the visual information with the audio stream using the discriminative traning process, and dispense with the multiple stream approach. Experiments are reported on the IBM infrared headset audio-visual database. On average of 20-speaker 1 hour speaker independent test data, the fMPE trained acoustic features achieve 33% relative gain. Adding video layers on top of audio layers gives additional 10% gain over fMPE trained features from the audio stream alone. The fMPE trained visual features achieve 14% relative gain, while the decision fusion of audio/visual streams with fMPE trained features achieves 29% relative gain. However, fMPE trained models do not improve over the original models on the mismatched noisy test data.

1. Introduction

Recently audio-visual speech recognition (AVSR) has attracted significant interest as a means of improving performance and robustness over audio-only speech recognition (ASR) [1, 2, 3], especially in real-life applications [4, 5]. The most successful AVSR systems extract visual features from the facial region of interest and combine them with acoustic features using multi-stream HMMs. It has been demonstrated that multi-stream decision fusion attains significant improvement in recognition accuracy over the single-stream based fusion methods [6].

Discriminative training techniques for HMM parameters can be naturally adapted to multi-stream HMM based AVSR system. Common discriminative training techniques such as MMI and MPE [7] have shown to give improvement on various tasks of ASR. Instead of discriminative training of HMM parameters, Li and Stern recently applied a discriminative-like normalized acoustic likelihood criterion to parallel feature generation [8]. The feature transforms of parallel data streams are estimated at the same time to maximize the normalized acoustic likelihood.

fMPE, introduced recently in [9], is also a feature space transform estimated discriminatively. It uses the Minimum Phone Error (MPE) discriminative criterion [7] to train a feature-level transformation. Unlike MPE training on HMM parameters, fMPE transforms training data with a kernel-like method and optimizes on large number of parameters comparable to the size of the acoustic model. Despite the large number of parameters, fMPE is robust to over-training. The fMPE transformation matrix projects posteriors of Gaussians to a feature space, and then adds the projected features to the original acoustic features. This matrix is trained from a zero start using a linear method. The improvement from fMPE is similar to MPE, around 10% relative [9] or more [10], and can be combined with model-space discriminative training such as MPE.

In this paper we investigate how fMPE trained features improve the performance of multi-stream HMM-based audio-visual speech recognition. In addition to estimating fMPE transforms individually for audio and visual streams, we take advantage of the flexible layer-based implementation of fMPE [10] to add visual information to the audio information using the fMPE training process. The resulting fMPE trained audio system achieves 10% additional gain over the fMPE trained audio system from audio information alone. Since visual information is noise invariant and added into the fMPE training for audio, the resulting system even improve a little on the mismatched noisy data compared to the baseline, while the fMPE trained audio system from audio alone degrades on the mismatched noisy data compared to the ML baseline.

The paper is organized as follows: the multi-stream HMM for AVSR is briefly reviewed in Section 2. Section 3 reviews fMPE first and then describes how to apply fMPE on combined visual and audio layers. The experimental setup and results are reported in Section 4, and conclusions are drawn in Section 5.

2. Multi-stream AVSR system

Our multi-stream HMM based AVSR system uses appearance-based visual features and decision fusion for the audio and visual streams. The visual features are extracted from the region of interest (ROI). We first estimate the location of the ROI, which contains the area around the speaker's mouth (see Section 4.1). Following ROI extraction, the visual features are computed by applying a two-dimensional separable DCT to the sub-image defined by the ROI, and retaining the top 100 coefficients with respect to energy. The resulting vectors then go though a pipeline consisting of intra-frame linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT), temporal interpolation, and feature mean normalization, producing a 30-dimensional feature stream at 100Hz. To account for inter-frame dynamics, fifteen consecutive frames in the stream are joined and subject to another LDA/MLLT step to give the final visual feature vectors with 41 dimensions [5].

In parallel to the visual feature extraction, audio features are also obtained, time synchronously, at 100 Hz. First, 24 mel frequency cepstral coefficients of the speech signal are computed over a sliding window of 25 msec, and are mean normalized to provide static features. Then, nine consecutive such frames

are concatenated and projected by means of LDA/MLLT onto a 60-dimensional space, producing dynamic audio features.

In the multi-stream HMM based decision fusion approach, the single-modality observations are assumed to be generated by audio-only and visual-only HMMs of identical topologies with class-conditional emission probabilities $P_a(\mathbf{o}_{a,t})$ and $P_v(\mathbf{o}_{v,t})$, respectively. Both are modeled as mixtures of Gaussian densities. Based on the assumption that audio and visual streams are independent, we compute the joint probability $P_{av}(\mathbf{o}_{av,t})$ as follows [11]:

$$P_{av}(\mathbf{o}_{av,t}) = P_a(\mathbf{o}_{a,t})^{\lambda} \times P_v(\mathbf{o}_{v,t})^{1-\lambda} \quad (1)$$

Exponent λ is used to appropriately weight the contribution of each stream, depending on the "relative confidence" on each modality. Exponents can be fixed or time dependent [12] (we use fixed weights). The use of stream exponents are critical to the robust operation of an AVSR system. Failure of either channel can be expected in any practical application, but the visual channel is much more prone to failure.

3. fMPE for audio-visual streams

We first briefly review the fMPE training process. Then we discuss how to use fMPE to directly add visual information to the audio features.

fMPE is a form of discriminative training that optimizes the same objective function as MPE, but does so by transforming the feature vectors. The MPE objective function is the average of the transcription accuracies of all possible sentences s, weighted by the probability of s given the model:

$$\mathcal{F}_{MPE}(\lambda) = \sum_{r=1}^{R} \sum_s P_\lambda^\kappa(s|\mathcal{O}_\nabla)\mathrm{A}(\int, \int_\nabla) \quad (2)$$

where $P_\lambda^\kappa(s|\mathcal{O}_\nabla)$ is defined as the scaled posterior sentence probability $\frac{p_\lambda(\mathcal{O}_\nabla|\int)^\kappa \mathcal{P}(\int)^\kappa}{\sum_u p_\lambda(\mathcal{O}_\nabla|\sqcap)^\kappa \mathcal{P}(\sqcap)^\kappa}$ of the hypothesized sentence s, where λ is the model parameters and $\mathcal{O}_\nabla$ the r'th sequence of acoustic data. The function $\mathrm{A}(s, s_r)$ is a "raw phone accuracy" of s given s_r, which equals the number of phones in the reference transcription s_r for file r, minus the number of phone errors.

Normally [7] the MPE objective function is used to train the acoustic model (means and variances); in fMPE it is use to train a feature transformation. In the original formulation of fMPE, the transformation is applied as follows:

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{M}\mathbf{h}_t, \quad (3)$$

where $\mathbf{x}_t$ are the original features on time t and $\mathbf{y}_t$ the modified features. $\mathbf{h}_t$ are high dimensional features calculated at each frame t, which may be a function of the original features $\mathbf{x}_t$. These are projected down with the matrix $\mathbf{M}$. If very high dimensions are used, it is important that the features $\mathbf{h}_t$ are sparse, i.e. very few of the elements of the vector are nonzero on each time frame. This means that very few of the rows of $\mathbf{M}$ have to be accessed on each time frame. The reason for adding the original features $\mathbf{x}_t$ is that it solves the problem of initializing the training algorithm with something reasonable. The matrix $\mathbf{M}$ can be trained from a zero start.

3.1. High-dimensional feature generation

The first stage of fMPE is to transform the features into a very high dimensional space. This is done as follows: a set of Gaussians is created by likelihood-based clustering of the Gaussians in the acoustic model to an appropriate size (300 in experiments reported here). On each frame, the Gaussian likelihoods are evaluated with no priors and a vector of posteriors. For the fMPE setup used here, the posteriors are augmented with the offset of the feature vector from each Gaussian's mean, multiplied by that Gaussian's posterior, to give a dimension in this case of $300(d+1)$, where d is the feature dimension. This is described more exactly in [10]. A key feature is that the vector $\mathbf{h}_t$ is sparse, i.e. only certain elements on each time t differ significantly from zero; this greatly speeds up the computation.

3.2. Acoustic context expansion

The vector is further expanded with left and right acoustic context. The following is a typical configuration used: If the central (current) frame is at position 0, vectors are appended which are the posterior vector at positions 1, and the average of the posterior vector at positions (2 and 3), (4 and 5) and (6, 7 and 8), and the same to the left (-1 etc), to give a vector nine times as large as the original one. This expansion process can be viewed as a matrix operation which itself is trained; see [10] for details.

3.3. Training the matrix

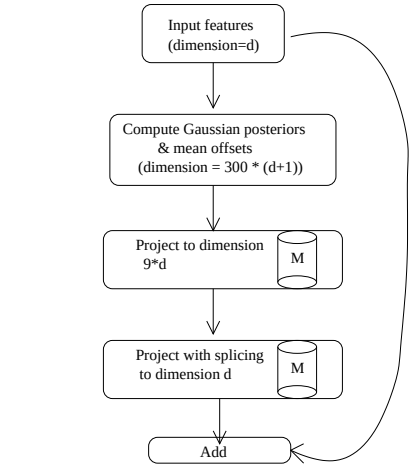
The matrix is trained by linear methods, because in such high dimensions accumulating squared statistics would be impractical. The training process is basically gradient descent, but various heuristics are used to give suitable learning rates for each matrix element, as described in [9, 10]. After each iteration of training the matrix, the recognition HMM is retrained for one iteration using normal E-M. However the Gaussians used to obtain the posteriors are kept constant. The process of obtaining the differential to update the matrix requires two passes over the data similar to normal discriminative training; the first pass is necessary to obtain differentials of the MPE criterion w.r.t. the HMM parameters which is used to give an extra term in the differential w.r.t. the matrix, a term which reflects the fact that the recognition HMM is going to be trained (with ML) using the same features.

3.4. Layered implementation

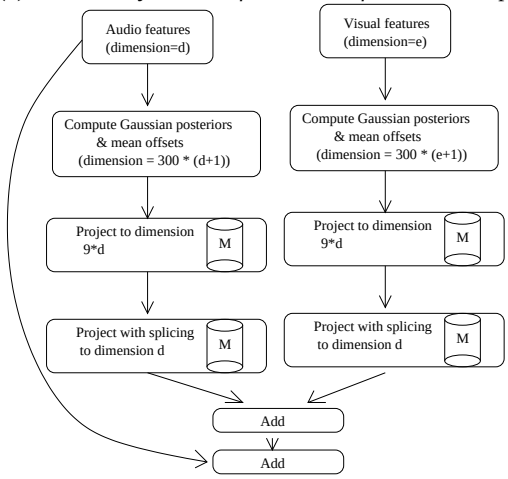
As described in [10], the calculation is now viewed as multiple layers of transformation which have a normalized interface. Instead of explicitly constructing the spliced vector $\mathbf{h}_t$ of size $9 \times 300 \times (d+1)$, we instead construct the un-spliced $\mathbf{h}_t$ of size $300 \times (d+1)$, project it to a dimension of $9d$ and then project to dimension d with a specialized form of frame splicing (which, to describe it briefly accomplishes projection in time but not in feature space).

Figure 3.4(a) shows the feature processing stages involved in the baseline fMPE setup. The layers with storage indicated in the diagram (a matrix "M") have trainable parameters; these are trained with a form of gradient descent. Some layers such as the layer that performs addition propagate the differentials back but have no trainable parameters. The layer that calculates the Gaussian posteriors is not trainable. Layers with very high dimensional outputs store their output with a sparse representation.

Figure 3.4(b) shows the audio-visual version of the fMPE training setup. The basic modules involved are the same, but some of them now have two instances and there is one more addition layer. The baseline features which are added to are the audio features, but both audio and video features are used to train offsets, which are simply added to each other. The layer-based setup means that no extra code was required to cope with



(a) Baseline layer-based feature transformation setup



(b) Audio-visual layer-based feature transformation setup

the feature combination.

4. Experimental setup and results

4.1. The infrared headset audio-visual database

Experiments are conducted on the audio-visual database collected with the IBM infrared headset [5]. The infrared headset is specially designed equipment that captures the video of the speaker's mouth region, independently of the speaker's movement and head pose. It reduces environmental lighting effect on captured images, allowing good visibility of the mouth ROI even in a dark room. Since the headset consistently focuses on the desired mouth region, face tracking is no longer required. Eliminating this step improves the visual front end robustness and reduces CPU requirements by approximately 40% [13].

The ROI extraction on headset captured videos is based on tracking the two mouth corners of the recorded subject. This corrects for slight positioning errors, boom rotation, and rescaling of the physical mouth size. Extracting the two mouth corners turns out to be fairly simple in the headset scenario, since it is assumed that the camera is already aimed nearly directly at the mouth. Because the types of features seen in the captured images are tightly constrained (i.e., no confusing background objects are expected in the scene), the algorithms can use very weak models and hence run quickly. The first step of the mouth finding algorithm estimates the position of the mouth in the image. The second step models the mouth as a dark slit and attempts to determine its corners. The final step is to output the normalized mouth image. The result is a 64×64 pixel ROI with an aspect ratio of about 1.7 covering the mouth. Details of the algorithms can be found in [5].

4.2. Experimental setup

Our AVSR system is built on 22kHz audio and 720x480 pixel resolution at 30 Hz video. A total of 107 subjects uttering approximately 35 random length connected digit sequences. We split 107 speakers in our infrared headset data into a training set and a testing set: 87 speakers are used for training, and the remaining 20 speakers are used for testing, and there is no overlap speakers in training and testing sets. The training data has about 4 hours of speech, and the test data has around 1 hour of speech. Both training and testing data have an average SNR of 20dB. In addition to this clean test data which matches the training data, another noisy test set is built by artificially corrupting the test set with additive "speech babble" noise resulting in an average SNR of 7dB. Recognition results are presented on both matched and noisy test sets.

The recognition system uses three-state, left-to-right phonetic HMMs with 159 context-dependent states (the context is cross-word, spanning up to 5 phones to either side) and $2,600$ Gaussian mixture components with diagonal covariances. At the decision fusion step, we keep the stream weights fixed, 0.7 for the audio stream, and 0.3 for the video stream. Since we don't have an extra validation test set, we fix the fMPE iteration number to 4 to avoid tuning the results on the test set.

To show the effectiveness of fMPE on multi-stream HMMs, we also present the fMPE results on individual single stream, audio-only, video+audio, and visual-only. The results are presented as word error rate (WER) for audio-only (A), video+audio combined with fMPE (V+A), visual-only (V) and multi-stream audio-visual (AV) recognition.

4.3. Results

Results are given before and after fMPE in Tables 1 and 2. Table 1 gives results on the single-stream HMMs; Table 2 gives results on multi-stream HMMs, i.e. combining as streams the individual systems shown in Table 1. The overall result seems to be that fMPE does very well on the conditions for which it was trained, i.e. in clean testing data, but the introduction of speech babble seems to generally eliminate or reverse the fMPE improvements.

The fMPE trained systems still give an improvement when tested in a multi-stream context (Table 2): the best result is combining as streams the fMPE trained audio and visual systems (AV) at 1.0%, which is a 29% relative improvement over the best non-fMPE system at 1.4% (AV). The combination of the (V+A) system which adds the visual information to an audio baseline using fMPE, with a visual (V) fMPE trained system ((V+A)V in Table 2) does not improve the (A+V) system much: 1.2% goes to 1.1%. But this is not surprising as the (A+V) system already contains the visual information.

One interesting pair of results to compare is the ML-trained AV result in Table 2, which uses streams to combine the two sources of information, at 1.4%, with the fMPE trained single stream (V+A) system in Table 1, at 1.2%. This means that fMPE can do a better job at combining the two sources of information than the multi-stream approach. However, we can do

	Match			Noisy	
	A	V+A	V	A	V+A
baseline	2.1	2.1	36.0	15.0	15.0
fMPE	1.4	1.2	30.9	18.7	14.6
rel. improvement	33%	43%	14%	-25%	3%

Table 1: *Comparison of single-stream fMPE results on audio-only, video+audio combined with fMPE, and visual-only speech recognition*

	Match		Noisy	
	AV	(V+A)V	AV	(V+A)V
baseline	1.4	1.4	8.8	8.8
fMPE	1.0	1.1	9.9	9.4
rel. improvement	29%	21%	-13%	-7%

Table 2: *Comparison of multi-stream fMPE results on audio-visual, and (video+audio)-visual speech recognition. XY means X and Y combined as streams*

better (1.0%) by training two separate fMPE systems and combining them using streams (AV in Table 2).

5. Conclusions

In this paper, we investigated the effect of fMPE training for digit recognition using audio and video features. fMPE training gave large improvements for both audio and video features, and gave even larger improvements for a system based on audio features when features calculated from the video stream was made available to the fMPE training process. Larger gains would have been possible from doing discriminative training (e.g. MPE) on top of the fMPE features, but we did not yet investigate this. We also intend to investigate combination with adaptation techniques such as fMLLR [14], which might increase the robustness of the fMPE trained system.

It was disappointing that the large gains from fMPE did not carry over to the mismatched test condition we investigated (additive speech babble), and to testing in a multi-stream context. This is surprising, especially since it was found in [15] that discriminative training works essentially as well across tasks as it does within tasks. Further investigation is necessary to determine whether is is a property of fMPE, of the specific fMPE features used, or whether it is an effect specific to this task. The strong confounding effect of the speech babble for this particular task has been observed in other conditions and it is probably dangerous to generalize too strongly from it. As regards the relatively small improvements of fMPE in the multi-stream case, it is perfectly possible in principle to train the fMPE transform when embedded in the multi-stream framework and this might give better results; however, this was not done as the current code does not support it.

It is encouraging that fMPE provides a way to take advantage of complementary information (visual features) without having to resort to a multi-stream approach. This could potentially lead to a more efficient system as we have to calculate fewer likelihoods in the acoustic model. It might also prove useful in an audio-only context, for example to integrate different types of features.

6. References

[1] Janin, A., Ellis, D., and Morgan, N., "Multi-stream speech recognition: Ready for prime time?", *Proc. Europ. Conf. Speech Technol.*, pp. 591–594, 1999.

[2] Dupont, S. and Luettin, J., "Audio-visual speech modeling for continuous speech recognition," *IEEE Trans. Multimedia*, 2(3): 141–151, 2000.

[3] Potamianos, G., Neti, C., Gravier, G., Garg, A., and Senior, A.W., "Recent advances in the automatic recognition of audio-visual speech," *Proc. IEEE*, 91(9): 1306–1326, 2003.

[4] G. Potamianos and C. Neti, "Audio-visual speech recognition in challenging environments," *Europ. Conf. Speech Commun. Technol.*, 2003.

[5] J. Huang, G. Potamianos, J. Connell and C. Neti, "Audio-Visual Speech Recognition Using an Infrared Headset," *Speech Communication*, Dec. 2004.

[6] E. Marcheret, S. Chu, V. Goel, G. Potamianos, "Efficient Likelihood Computation in Multi-Stream HMM Based Audio-Visual Speech Recognition," *Int. Conf. Speech and Language Processing*, 2004.

[7] D. Povey and P. C. Woodland, "Minimum Phone Error and I-smoothing for Improved Discriminative Training," *ICASSP*, 2002.

[8] L. Xiang and R. M. Stern, "Parallel Feature Generation Based on Maximizing Normalized Acoustic Likelihood," *ICSLP*, 2004.

[9] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, G. Zweig, "fMPE: Discriminatively trained features for speech recognition," *ICASSP*, 2005.

[10] D. Povey, "Improvements to fMPE for discriminative training of features," submitted to *Interspeech* 2005.

[11] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," IEEE Trans. Multimedia, 2(3):141–151, 2000.

[12] A. Garg, G. Potamianos, C. Neti, T. Huang, "Frame-Dependent Multi-Stream Reliability Indicators for Audio-Visual Speech Recognition," *Int. Conf. Acoustic Speech and Signal Processing*, 2003.

[13] J. Connell, N. Haas, E. Marcheret, C. Neti, G. Potamianos, S. Velipasalar, "A Real-Time Prototype for Small-Vocabulary Audio-Visual ASR," *IEEE Int. Conf. on Multimedia & Expo*, 2003.

[14] J. Huang, E. Marcheret, K. Visweswariah, "Rapid Feature Space Speaker Adaptation for Multi-Stream HMM-based Audio-Visual Speech Recognition," *IEEE Int. Conf. on Multimedia & Expo*, 2005, to appear.

[15] R. Cordoba, P.C. Woodland & M.J.F. Gales, "Improving cross-task performance using MMI training," *Int. Conf. Acoustic Speech and Signal Processing*, 2002.