Combined Discriminative Training for Multi-Stream HMM-based Audio-Visual Speech Recognition

Jing Huang\textsuperscript{1}, Karthik Visweswariah\textsuperscript{2}

\textsuperscript{1}IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
\textsuperscript{2}IBM India Research Lab
Bangalore, India

\texttt{jghg@us.ibm.com, v-karthik@in.ibm.com}

Abstract

In this paper we investigate discriminative training of models and feature space for a multi-stream hidden Markov model (HMM) based audio-visual speech recognizer (AVSR). Since the two streams are used together in decoding, we propose to train the parameters of the two streams jointly. This is in contrast to prior work which has considered discriminative training of parameters in each stream independent of the other. In experiments on a 20-speaker one-hour speaker independent test set, we obtain 22\% relative gain on AVSR performance over A/V models whose parameters are trained separately, and 50\% relative gain on AVSR over the baseline maximum-likelihood models. On a noisy (mismatched to training) test set, we obtain 21\% relative gain over A/V models whose parameters are trained separately. This represents 30\% relative improvement over the maximum-likelihood baseline.

Index Terms: discriminative training, audio-visual speech recognition, multi-stream HMM

1. Introduction

Audio-visual speech recognition (AVSR) has attracted significant interest as a means of improving performance and robustness over audio-only speech recognition \cite{1, 2, 3}, especially in noisy environments \cite{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 hidden Markov models (HMMs).

However, in practical scenarios, where the subject’s posture and the environment lighting are hard to control (e.g. in automobiles and offices), robust extraction of the visual speech information becomes a rather challenging problem. It requires accurate tracking of the speaker’s face and facial features (e.g. mouth corners and lip contours), as well as successful compensation for head pose and lighting variations.

These facts motivated various efforts to improve the robustness of the visual stream. In \cite{7} visual features are adapted using feature space maximum likelihood linear transforms (fMLLR) to improve the visual stream in realistic environments. fMLLR was shown to improve visual recognition word error rate (WER) from 36\% to 22\% on continuous variable-length digit recognition task, and adding a global non-linear Gaussianization transform on top of fMLLR improved WER further to 20\%. In \cite{8} a multi-objective optimization function was proposed to discriminatively train the visual HMM model. On an isolated Korean digit task their objective function was shown to be better than minimum classification error (MCE) and maximum mutual information (MMI) objective functions. Although these efforts improve the performance based on the visual channel alone this is not enough to improve the overall ASR performance, as was pointed out in the conclusion of \cite{7}. Ideally we would like the information in the visual stream to complement the audio stream so that the overall AVSR performance improves.

Another focus for improving speech recognition systems has been to focus on discriminative training of parameters using techniques such as MMI, minimum phone error (MPE) \cite{9} and MPE trained features (fMPE) \cite{10} that have shown consistent improvements on various speech recognition tasks. These techniques can be naturally extended to multi-stream HMM based AVSR systems by applying these methods to each stream independently. In \cite{11} fMPE training was investigated for a multi-stream HMM AVSR system. It was shown that fMPE trained individually for each stream gave large improvements, 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. However, it was disappointing that the large gains from fMPE did not carry over to mismatched test conditions.

The focus of this paper is to train the parameters in each of the streams (audio and visual) to optimize the performance of the overall AVSR system. We do this in a discriminative training framework using the MPE criterion (although other discriminative criteria could be used as well). Since the gap between recognition accuracy from the audio and visual channels is huge (for example, for continuous digit recognition task, audio accuracy is close to 98\%, while visual accuracy is less than 70\%), the audio stream dominates the AVSR performance. The visual stream serves as a complementary helper when the audio stream is weak (for example in noisy conditions). This motivated us to focus on training the parameters of the visual stream (model and feature transform parameters) while trying to optimize the video stream to optimize the overall performance.

We show in Section 6 that our proposed way of combined audio-visual MPE/fMPE training indeed improves AVSR performance over that of MPE/fMPE models trained independently on the audio and visual streams. This improvement is achieved in both matched and mismatched conditions. We also investigate the effect of using feature space adaptation (fMLLR) estimated from multi-stream HMMs applied on top of MPE/fMPE models. We show that the gains obtained from joint training carry over on top of the gains obtained by fMLLR adaptation.

The rest of the paper is organized as follows: our multi-stream HMM system for AVSR is briefly reviewed in Section 2.
Section 3 and Section 4 review MPE and fMPE (respectively) and then describe how to train visual MPE/fMPE using combined audio-visual information. Section 5 briefly describes the application of fMLLR to our AVSR system. The experimental setup and results are reported in Section 6, and conclusions are drawn in Section 7.

2. Multi-stream HMM-based AVSR system

Our AVSR system is an HMM-based speech recognizer, with appearance-based visual features and decision fusion for the audio and visual streams (referred to as multi-stream HMMs).

The visual features are computed using a discrete Cosine transform applied to the sub-image defined by the region of interest. We then perform various processing steps to retain the most useful information, to ensure the feature components are uncorrelated and to match the sampling rate of the audio features. The reader is referred to [5] for details of the feature extraction.

The audio feature extraction is fairly standard: we use mean normalized mel frequency cepstral coefficients (calculated using a sliding window of 25ms) followed by linear discriminant analysis (LDA) and maximum likelihood linear transforms (MLLT) resulting in a 60-dimensional feature space.

In the multi-stream HMM decision fusion approach, the single-modality observations are assumed generated by audio-only and visual-only HMMs of identical topologies with class-conditional emission probabilities $P_c(a,t | c)$ and $P_c(o,v,t | c)$, respectively, where $c \in C$ denotes the speech classes of interest such as context-dependent sub-phonetic units. Both are modeled as mixtures of Gaussian densities. Based on the assumption that audio and visual streams are independent, we compute the joint probability $P_{av}(o_{av},t | c)$ as follows [2]:

$$P_{av}(o_{av},t | c) = P_{a}(o_{a},t | c) \times P_{v}(o_{v},t | c)^{1-\alpha}$$  \hspace{1cm} (1)

Exponent $\alpha$ is used to appropriately weight the contribution of each stream, depending on the "relative confidence" of each modality. Failure of either channel can be expected in any practical application, the visual channel is much more prone to failure. Although time-dependent $\alpha$ could be used to compensate for these failures, we use a fixed $\alpha$ value.

3. MPE training for multi-stream AVSR

Throughout this paper we use the MPE objective function (specifically we use the MPE objective function given in [12]) which is the average of the frame phone accuracies of all possible sentences $s$, weighted by the posterior probability of $s$ given the model and observations:

$$F_{MPE}(\lambda) = \frac{1}{R} \sum_{r=1}^{R} \sum_{s} P_{a}(s|O|) \Lambda(s, s_r)$$ \hspace{1cm} (2)

where $P_{a}(s|O)$ is the posterior sentence probability of the hypothesized sentence $s$, where $\Lambda(s, s_r)$ is a frame level phone accuracy of a hypothesis $s$ given the reference $s_r$, which was introduced in [12].

To adapt MPE training to audio-visual AVSR system, a natural way to train the models for the audio and visual stream independently of each other, i.e., we train the acoustic model $\lambda_a$ to optimize $F_{MPE}(\lambda_a)$ and similarly for the visual stream.

Since in the decoding process the two features streams and models are used together we instead propose to train both the models jointly by optimizing:

$$F_{MPE}(\lambda_a, \lambda_v) = \sum_{r=1}^{R} \sum_{s} P_{a}(s|O_a) \sum_{c} P_{v}(c|O_v) \Lambda(s, s_r)$$  \hspace{1cm} (3)

In calculating the scaled sentence posterior probabilities we use a lattice generated by the joint audio-visual model and use the joint model in all calculations on the lattice. Thus, we train parameters in each of the streams to complement the other stream.

The standard approach to optimizing the MPE objective function $F_{MPE}(\lambda_a)$, given by [13], calculates

$$\gamma_d^{MPE}(t) = \frac{1}{k} \frac{\partial F_{MPE}}{\partial \log p(o_{av}, t | q)}$$ \hspace{1cm} (4)

which is then used to update numerator statistics (if it is positive) or denominator statistics (if it is negative). In Equation 4, $q$ is a context-dependent HMM state. These statistics are then used to update the model parameters using the standard EBW parameter update equations. To do parameter updates of the joint objective function in Equation 3, the only change we need to make is to calculate $\gamma_{MPE}(t)$ using the audio-visual model operating on the audio-visual features and using a lattice generated with an audio-visual system. We then gather statistics and perform updates on the audio and/or visual stream in the standard manner.

4. fMPE training for multi-stream AVSR

fMPE is a form of discriminative training that optimizes the same objective function as model MPE, but does so by transforming the feature vectors as follows:

$$y_t = o_t + M y_t$$ \hspace{1cm} (5)

where $o_t$ are the original features on time $t$ and $y_t$ the modified features. $h_i$ are high dimensional features calculated at each frame $t$, which are projected down with a matrix $M$. Matrix $M$ can be trained from a zero start (details see [13]).

In training the fMPE transform for audio-visual systems, previous work [11] simply optimized the fMPE parameters independently for each stream. Lattices were generated separately for the audio and visual streams, and for each feature stream the gradient of the MPE objective function was calculated with respect to the features in that stream.

In contrast, we consider the joint objective function given in Equation 3 (now viewed as a function of the fMPE transform parameters for the audio ($\lambda_a$) and visual streams ($\lambda_v$)). To evaluate the joint MPE objective a single set of lattices are generated with joint audio-visual model. As in [13], we update parameter using a gradient update. We expand the partial derivative of the MPE objective function with respect to, say, the visual feature stream, as:

$$\frac{\partial F_{MPE}}{\partial o_{av,t}} = \sum_{q} \frac{\partial F_{MPE}}{\partial \log p(o_{av,t} | q)} \frac{\partial \log p(o_{av,t} | q)}{\partial o_{av,t}}$$

The second term on the R.H.S above is further split up into direct and indirect differentials (see [13]), and only involves the video stream. The first term (which is the same as the term required in model MPE update) involves calculations using both the audio and visual streams and uses lattices generated by an
audio-visual combined decoder. Although in principle we can jointly update audio and visual fMPE parameters, due to time limitation we only implemented training of visual fMPE parameters using the joint procedure described above with audio fMPE model trained using the standard fMPE training procedure.

5. fMLLR for multi-stream AVSR

fMLLR is a widely used and effective adaptation technique for reducing the mismatch between training and test conditions. In fMLLR the feature \( \mathbf{o} \) is transformed linearly to maximize the likelihood of the testing data. In multi-stream AVSR application, fMLLR transforms are estimated as follows [14]: let \( \mathbf{o}_a \) denote the audio feature vector and \( \mathbf{o}_v \) denote the video feature vector.

\[
\begin{pmatrix} \mathbf{y}_a \\ \mathbf{y}_v \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{O} \\ \mathbf{O} & \mathbf{V} \end{pmatrix} \begin{pmatrix} \mathbf{o}_a \\ \mathbf{o}_v \end{pmatrix} + \begin{pmatrix} \mathbf{b}_a \\ \mathbf{b}_v \end{pmatrix} \tag{6}
\]

where \( \mathbf{A} \) and \( \mathbf{V} \) are fMLLR transformation matrices for audio and visual stream respectively, and \( \mathbf{O} \) is zero matrix. In standard single stream fMLLR (the top block of (6) expressed as \( \mathbf{y} = \mathbf{Ao} + \mathbf{b} \)) the objective function is [15]

\[
Q(\mathbf{W}) = \log |\det \mathbf{A}| - 1/2 \sum \mathbf{w}_i^T \mathbf{G}_i \mathbf{w}_i + k_i \mathbf{w}_i^T \mathbf{w}_i \tag{7}
\]

where the mean and variance statistics \( k_i \) and \( \mathbf{G}_i \) respectively are gathered from the adaptation data and \( \mathbf{w}_i = [\mathbf{a}_ib_i] \) is a vector made of the \( i \)th row of the transform \( \mathbf{A} \) and the \( i \)th element of \( \mathbf{b} \). In the case of multi-stream HMMs we have the same objective function except the statistics \( \mathbf{G}_i \) and \( k_i \) are gathered with posterior calculated jointly using audio and visual streams.

6. Experimental setup and results

6.1. Experimental setup

Experiments are conducted on the audio-visual database collected with the IBM infrared headset [5]. Our AVSR system is built on 16kHz audio and 720x480 pixel resolution at 30 Hz video. The data consists of a total of 107 subjects uttering approximately 35 random length connected digit sequences. We split the 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 in speakers in the training and testing sets. The training data has about 4 hours of speech, and the test data has around 1 hour of speech with around 6300 digit words. Both training and testing data have an average SNR of 20dB. In addition to this clean test data which matches the training data, a noisy test set is built with 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 mismatched noisy test sets. The mismatched conditions are significantly more challenging and in the past gains from discriminative training have not carried over to the mismatched conditions [11].

The baseline 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, 0.8/0.7 for the clean/noisy audio stream respectively. Since we don’t have an extra validation set, we fix the MPE/fMPE training iteration number to 4 to avoid tuning the results on the test set. Note that the only difference in data between [11] and this one is 22KHz audio vs. 16KHz audio format. We used different sizes of audio/visual models in [14].

<table>
<thead>
<tr>
<th>System</th>
<th>Match</th>
<th>Noisy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>V</td>
</tr>
<tr>
<td>baseline</td>
<td>34.1</td>
<td>1.4</td>
</tr>
<tr>
<td>MPE</td>
<td>33.5</td>
<td>0.9</td>
</tr>
<tr>
<td>MPE_VC</td>
<td>36.5</td>
<td>0.8</td>
</tr>
<tr>
<td>MPE_J</td>
<td>38.5</td>
<td>0.7</td>
</tr>
<tr>
<td>fMPE</td>
<td>30.0</td>
<td>1.1</td>
</tr>
<tr>
<td>fMPE_VC</td>
<td>39.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 1: Comparison of baseline, various MPE and fMPE models.

6.2. Results

Results are presented as word error rate (WER) for audio-only (A), visual-only (V) and multi-stream audio-visual (AV) recognition. MPE_AV refers to A/V models trained separately; MPE_VC refers to A/V models trained as follows: train MPE A model first, then use MPE A model combined with V model (initially maximum-likelihood trained) to train the MPE V model. MPE_J refers to MPE models trained jointly for A and V. Same notations refer to corresponding fMPE models (though fMPE_J is not done due to time constraint).

Tables 1 gives comparison of baseline models and various trained MPE/fMPE models. On the matched test data, first noticed is the huge relative gain of 50% from MPE A model compared to the baseline A model. By comparison MPE V model only shows tiny gain over the baseline V model. In total there is still 36% gain for AV performance from MPE models. In comparison, MPE_VC trained V model degrades 7% from the baseline V model. However, since it is trained combined with MPE A model, the overall AV performance improves 43% over baseline, 11% more on top of the MPE separately trained A/V models. The best results come from MPE_J; trained V model degrades even more, i.e. 13% from the baseline V model. However, since it is trained jointly with A model, the overall AV performance improves 50% over baseline, 22% more on top of the MPE separately trained A/V models.

On the mismatched noisy condition, MPE_A model does not help, in fact degrades a little (7%) from the baseline. Overall the MPE AV model performance stays the same as the baseline maximum likelihood system performance. However, MPE_VC and MPE_J still shows 8% and 9% relative improvement on the MPE AV performance, even though both A/V degrade 7% and 13% from the baselines. This suggests relative robustness of jointly trained AV MPE models on mismatched test data.

For fMPE models there is reasonable relative gain on the matched test data: 40% for audio-only, and more than 20% for AV. There is 15% degradation from fMPE_VC V model, however, fMPE_VC models still adds additional 9% on top of fMPE models for the AV performance.

Overall fMPE is not as good as MPE. In fact, when audio fMPE is followed by MPE, the performance on audio-only is still not as good as MPE A model (1.1% vs. 1.0%). Thus we did not report any other results on fMPE+MPE. The gains from MPE do not carry over to the mismatched test condition, however, while MPE models have much better results on the mismatched test data, with over 8% relative improvement over the baseline AV number. Since discriminatively trained models seem to perform relatively poorly in mismatched conditions, we investigate how fMLLR adaptation can compensate for the mismatch and boost the performance of discriminatively trained models on mismatched test data.

There are two ways of adapting MPE/fMPE models us-
In this paper we propose an effective way to jointly train audio-visual models and feature transforms discriminatively using the MPE objective function. We have shown how to incorporate combined AV models in the process of MPE/IMPE training. We show from experimental results that this approach is better than training audio and visual stream models and feature transforms separately. Although the visual-only performance degrades using this approach of training, the overall multi-stream AVSR performance improves significantly in all cases. On the matched test conditions, we obtain 50%/30% relative gain from joint training MPE/IMPE systems relative to the maximum-likelihood AVSR baseline. This represents a gain of 22%/9% relative to AV MPE/IMPE models trained independently on each stream. On the mismatched noisy test data, we adapt the discriminative models using multi-stream feature-space maximum likelihood linear regression (fMLLR) to compensate for some of the mismatch. Again, our jointly trained MPE/IMPE systems achieve significant gains. We get 30%/19% gains using joint training for MPE/IMPE systems relative to baseline fMLLR adapted AV models, which is a gain of 21%/9% relative to the corresponding models with the streams trained independently.

It is encouraging that our proposed combined AV trained MPE models perform very well on matched test data as well as mismatched test data, although IMPE models still suffer the problem of generalization as observed in [11]. We will implement combined AV fMPE training and see whether this problem still persists. Since these results are obtained on a relatively small task, we are currently experimenting on a 50-hour dataset for large-vocabulary speech recognition task with the same infrared headset.

### 8. References


