

Efficient Likelihood Computation in Multi-Stream HMM based Audio-Visual Speech Recognition

Etienne Marcheret, Stephen M. Chu, Vaibhava Goel, Gerasimos Potamianos

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

{etiennem,schu,vgoel,gpotam}@us.ibm.com

Abstract

Multi-stream hidden Markov models have recently been introduced in the field of automatic speech recognition as an alternative to single-stream modeling of sequences of speech informative features. In particular, they have been very successful in audio-visual speech recognition, where features extracted from video of the speaker’s lips are also available. However, in contrast to single-stream modeling, their use during decoding becomes computationally intensive, as it requires calculating class-conditional likelihoods of the added stream observations. In this paper, we propose a technique to reduce this overhead by drastically limiting the number of observation probabilities computed for the visual stream. The algorithm estimates a joint co-occurrence mapping of the Gaussian mixture components that separately model the audio and visual observations, and uses it to select the visual mixture components to be evaluated, given the already selected audio ones. We report experiments using this scheme on a connected-digits audio-visual database, where we demonstrate significant speed gains at decoding with only about 5% of the visual Gaussian components requiring evaluation, as compared to the independent evaluation of audio and visual likelihoods.

1. Introduction

Recently, there has been significant interest in the use of multi-stream hidden Markov models (HMMs) for automatic speech recognition (ASR) [1]. For example, such models have been successfully considered for multi-band ASR [2], separate static and dynamic acoustic feature modeling [3], as well as for audio-visual ASR [4], [5].

In its application in audio-visual speech recognition, the multi-stream approach gives rise to an effective paradigm to fuse and model the two separate information sources carried in the audio and visual observations. Specifically, it has been demonstrated that multi-stream decision fusion attains significant improvement in recognition accuracy over the state-of-the-art single-stream based fusion methods, e.g., hierarchical linear discriminant analysis (HiLDA) [5].

However, the gain in recognition performance is achieved at the cost of higher computational complexity due to the separate statistical modeling of the two observation streams. For instance, in the audio-visual ASR system described in [5], the signal processing front end produces audio and visual observation vectors with 60 and 41 dimensions, respectively. In HiLDA fusion, the joint audio-visual observations of 101 dimensions are projected to a 60 dimensional audio-visual feature space, which can be modeled by single-stream HMMs with similar number of Gaussian densities as the audio only system. On the other hand, the multi-stream HMMs model each of the two modalities in its original feature space. Hence, the number of Gaussian components required is roughly doubled in order to preserve the same modeling resolution in the output densities. For a typical decoding algorithm, the time complexity is roughly linear with

respect to the total number of Gaussians in the system. Therefore, without special treatment, an audio-visual system based on two-stream HMMs will approximately command twice the computational load as a comparable single-stream system in the recognition stage.

Effectively managing the computational load is essential to the development of real-time audio-visual ASR systems [6]. Because visual processing is expected to take a sizeable portion of the available computing power, it becomes even more imperative to improve the efficiency of algorithms involved in the decoding process, which include likelihood computation and search. In this work, we shall concentrate on the former, and develop an efficient algorithm to evaluate the mixtures of Gaussian densities in multi-stream HMMs.

Algorithms exist for fast evaluation of Gaussians in single-stream HMMs. One class of algorithms exploits the fact that at a given frame, only a small subset of Gaussian components in the total Gaussian pool are significant to the likelihood computations, e.g., the roadmap algorithm and the hierarchical labeling algorithm [7]. Naturally, these algorithms may be directly applied to the each individual stream in the multi-stream HMM. Moreover, the synchronized and parallel nature of the observation streams in multi-stream HMMs provides a fresh dimension to formulate new approaches to further improve computational efficiency. To our knowledge, no existing technique has explored this direction to date.

In this paper, we describe a novel algorithm that estimates a co-occurrence mapping of the Gaussian mixture components that separately model individual streams of a multi-stream system. The method essentially treats stream pairs in a master/slave fashion, with the master Gaussian components driving the slave component selection. We find that on an audio-visual digit recognition task the algorithm can achieve significant improvement in decoding efficiency with a minimal degradation in recognition performance.

The rest of the paper is organized as follows. In Section 2, we describe the hierarchical labeling algorithm for single-stream HMMs; Section 3 gives the details of the Gaussian co-occurrence algorithm; the audio-visual ASR system is described in Section 4; the experimental results are presented in Section 5, and we conclude the paper in Section 6.

2. Hierarchical Labeling Algorithm

In an HMM, the emission density function of a state is typically parameterized by a mixture of Gaussian densities. The state conditional likelihood of a given observation vector \mathbf{x} at time t is computed as

$$p(\mathbf{x}|s) = \sum_{g \in \mathcal{G}(s)} p(g|s)p(\mathbf{x}|g) \quad (1)$$

where $\mathcal{G}(s)$ is the set of Gaussians that make up the GMM for state s .

The evaluation of a Gaussian density can be carried out on-demand as a state associated with the particular Gaussian is invoked; or, alternatively, a set of Gaussians can be precomputed as soon as the observation is available without regard to their state membership. We shall refer the former as the *lazy* method, and the latter as the *eager* method. It is apparent that for a system with a large number of Gaussians, only a small subset of the complete set of Gaussian densities are significant to likelihood computation at any given time. Hence, clever exploitation of this sparseness combined with the *eager* method yields a very efficient algorithm to compute the conditional likelihoods during decoding.

The hierarchical labeling algorithm takes advantage of the sparseness by surveying the Gaussian pool in multiple resolutions given a feature vector \mathbf{x} . As a part of the training process, the complete set of available Gaussian densities is clustered into a search tree, in which the leaves correspond to the individual Gaussians, and a parent node is the centroid of its children. Thus, levels closer to the root node can be viewed as lower resolution representations of the feature space. In the experiments described in this paper, the trees have four levels.

During decoding, for each feature frame, the tree is traversed to identify a subset of active Gaussians, \mathcal{Y} . Based on \mathcal{Y} , the conditional likelihood of a state is computed using the following approximation

$$p(\mathbf{x}|s) = \max_{g \in \mathcal{Y} \cap \mathcal{G}(s)} p(g|s)p(\mathbf{x}|g) \quad (2)$$

If no Gaussian from a state is present in \mathcal{Y} , a default floor likelihood is assigned to that state.

3. Using Gaussian Co-Occurrence

The hierarchical labeling algorithm described in the previous section relies on the hierarchical tree to give a list of active Gaussian densities for the current observation. A straightforward application of the algorithm to multi-stream HMMs is to consider a separate tree for each stream and determine the active Gaussians in independence from other streams. However, even with the highly efficient pruning provided by hierarchical labeling, the task of Gaussian computation still accounts for approximately 50% to 70% of the total recognition effort in our single-stream ASR system. So, the independent hierarchical labeling scheme is unsuitable for realtime implementation of multi-stream HMMs.

The synchronous, parallel streams in a multi-stream HMM are typically used to model different aspects of the same underlying stochastic process. Therefore, it is natural to hypothesize that some degrees of inter-stream dependencies exist among the feature spaces. Indeed, this conjecture directly leads to the formulation of Gaussian co-occurrence modeling. Particularly, we propose to apply hierarchical labeling in only one of the streams, and use co-occurrence statistics to determine the active Gaussian components for the rest of the streams.

To simplify discussion, we shall restrict the subsequent derivations to the two-stream case. However, note that the formulation is completely general, and the equations can be readily extended to include more than two observation streams.

Given feature vectors from two streams, \mathbf{x}_1 from stream1 and \mathbf{x}_2 from stream2, we wish to compute the joint probability $p(\mathbf{x}_1, \mathbf{x}_2|s)$ for HMM state s . Multi-stream systems typically make the assumption that, conditioned on HMM state, the streams are independent [5]. Consequently, the joint probability

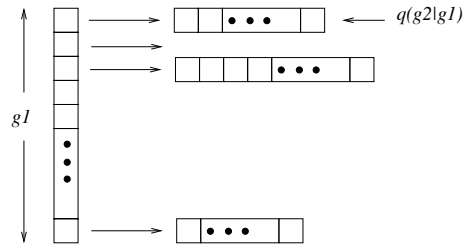


Figure 1: Gaussian co-occurrence map.

is factored as

$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2|s) &= p(\mathbf{x}_1|s)p(\mathbf{x}_2|s) \\ &= \left\{ \sum_{g_1 \in \mathcal{G}_1(s)} p(g_1|s)p(\mathbf{x}_1|g_1) \right\} \times \\ &\quad \left\{ \sum_{g_2 \in \mathcal{G}_2(s)} p(g_2|s)p(\mathbf{x}_2|g_2) \right\}. \end{aligned} \quad (3)$$

Under hierarchical labeling (equation 2), equation 3 is approximated as

$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2|s) &= \left\{ \max_{g_1 \in \mathcal{Y}_1 \cap \mathcal{G}_1(s)} p(g_1|s)p(\mathbf{x}_1|g_1) \right\} \times \\ &\quad \left\{ \max_{g_2 \in \mathcal{Y}_2 \cap \mathcal{G}_2(s)} p(g_2|s)p(\mathbf{x}_2|g_2) \right\}, \end{aligned} \quad (4)$$

where \mathcal{Y}_1 and \mathcal{Y}_2 are the Gaussians resulting from the hierarchical labeling of stream1 and stream2, respectively.

In the Gaussian co-occurrence method we attempt to model the inter-stream dependence. We start by removing the independence assumption made in equation 3, and we rewrite the state conditional likelihood as

$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2|s) &= \sum_{g_1, g_2 \in \mathcal{H}} p(\mathbf{x}_1, g_1, \mathbf{x}_2, g_2|s) \\ &= \sum_{g_1, g_2 \in \mathcal{H}} p(\mathbf{x}_1, g_1|s)p(\mathbf{x}_2, g_2|\mathbf{x}_1, g_1, s), \end{aligned} \quad (5)$$

where \mathcal{H} is the set of all Gaussians belonging to HMMs used for modeling the two streams. We note that in our system the Gaussians are not shared across states or streams, and hence, for any given state, only Gaussians belonging to that state will be effective in the summation.

Let $\mathcal{Q}_1 \subseteq \mathcal{Y}_1$ be a set of stream1 Gaussians. The details of how \mathcal{Q}_1 is determined are discussed in section 5.2. Using \mathcal{Q}_1 we approximate the second term in equation 5 as

$$p(\mathbf{x}_2, g_2|\mathbf{x}_1, g_1, s) \approx p(\mathbf{x}_2, g_2|\mathbf{x}_1, \mathcal{Q}_1, s). \quad (6)$$

We then further approximate the RHS of equation 6 as

$$\begin{cases} p(\mathbf{x}_2, g_2|\mathbf{x}_1, \mathcal{Q}_1, s) \approx \\ \quad p(\mathbf{x}_2, g_2|s), \text{ if } \max_{g'_1 \in \mathcal{Q}_1} q(g'_1, g_2|\mathbf{x}_1) > t_1; \\ 0, \text{ otherwise.} \end{cases} \quad (7)$$

In equation 7, $q(g'_1, g_2|\mathbf{x}_1)$ denotes a distribution modeling the joint occurrence of Gaussians of stream1 and stream2 and t_1 is an empirically determined threshold.

Equation 7 in essence uses the co-occurrence distribution $q(g'_1, g_2|\mathbf{x}_1)$ and set \mathcal{Q}_1 to limit the number of Gaussians of stream2 that are evaluated. Let

$$\mathcal{Q}_2 = \{g_2 : \max_{g'_1 \in \mathcal{Q}_1} q(g'_1, g_2|\mathbf{x}_1) > t_1\} \quad (8)$$

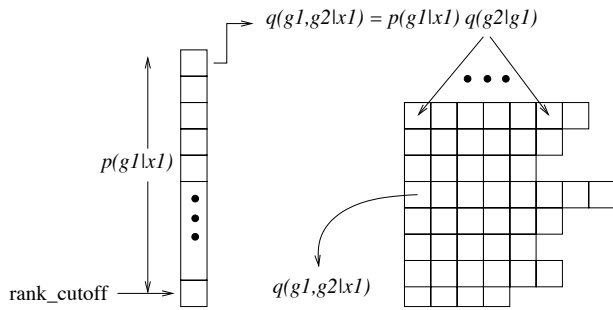


Figure 2: Run-time use of co-occurrence map, generating the co-occurrence distribution $q(g_1, g_2|\mathbf{x}_1)$.

denote the set of Gaussians that are evaluated for stream2. Controlling the size of \mathcal{Q}_2 is how we expect to derive primary computational savings. Note that we are using distribution q to only determine \mathcal{Q}_2 ; in the future we plan to investigate the use of the probability values from this distribution as well.

Combining equations 5 and 7, and using the max approximation to summation, as was done in equation 2, we get

$$p(\mathbf{x}_1, \mathbf{x}_2|s) = \left\{ \max_{g_1 \in \mathcal{D}_1 \cap \mathcal{G}_1(s)} p(g_1|s)p(\mathbf{x}_1|g_1) \right\} \times \left\{ \max_{g_2 \in \mathcal{Q}_2 \cap \mathcal{G}_2(s)} p(g_2|s)p(\mathbf{x}_2|g_2) \right\}. \quad (9)$$

As it becomes apparent from equation 8, the distribution $q(g_1, g_2|\mathbf{x}_1)$ plays a central role in the amount of computational savings that we can derive from this method. We model this distribution as

$$q(g_1, g_2|\mathbf{x}_1) = P_{\mathcal{Q}_1}(g_1|\mathbf{x}_1)q(g_2|g_1), \quad (10)$$

where $P_{\mathcal{Q}_1}(g_1|\mathbf{x}_1)$ is computed at test time from the likelihoods $p(\mathbf{x}_1, g_1)$ given by hierarchical labeling of stream1, as

$$P_{\mathcal{Q}_1}(g_1|\mathbf{x}_1) = \frac{p(\mathbf{x}_1, g_1)}{\sum_{g'_1 \in \mathcal{Q}_1} p(\mathbf{x}_1, g'_1)}. \quad (11)$$

The conditional distribution $q(g_2|g_1)$ is computed at training time by “counting” the instances where g_1 occurs in stream1 together with g_2 in stream2. Specifically, it is derived from the empirical expectation

$$q(g_1, g_2) = \frac{1}{|T|} \sum_t p(g_1|\mathbf{x}_1(t))p(g_2|\mathbf{x}_2(t)), \quad (12)$$

where $|T|$ is the total number of training feature vectors.

For storage efficiency, we sort $q(g_2|g_1)$ in descending order and store only top few g_2 Gaussians for each g_1 . This stored map is referred to as the Gaussian co-occurrence map. Figure 1 gives a graphical rendering of a Gaussian co-occurrence map.

At test time, the Gaussian co-occurrence map is used in conjunction with $P_{\mathcal{Q}_1}(g_1|\mathbf{x}_1)$ values, computed according to equation 11. Figure 2 shows the use of the co-occurrence map at runtime.

4. Multi-Stream Audio-Visual Speech Recognition System

Our multi-stream configuration consists of three streams: an audio stream (AU), a visual stream (VI), and an audio visual fused stream (AVf), implemented using the HiLDA approach [5].

The visual front-end in the audio-visual speech recognition system extracts appearance-based features within a region of interest (ROI) defined on the mouth area of the speaker.

| SNR | 19.5dB | 11.5dB | 8.5dB |
|-----------------|--------|--------|-------|
| AU | 1.60 | 13.45 | 25.78 |
| AVf | 1.65 | 9.38 | 15.98 |
| VI | 37.13 | 37.13 | 37.13 |
| AVf + VI (Ind.) | 1.59 | 7.85 | 12.12 |
| AU + VI (Ind.) | 1.61 | 8.97 | 14.10 |
| AVf + VI (Co.) | 1.61 | 7.62 | 12.06 |
| AU + VI (Co.) | 1.46 | 9.57 | 16.03 |

Table 1: Word error rates for single- and multi-stream independent, and co-occurrence systems.

Given the video input, the system first performs face detection at frame-level, using multi-scale template matching based on a distance measure composed of the two-class (face/non-face) Fisher linear discriminant and the error incurred by projecting the candidate vector to a lower dimensional “face space” obtained through principal component analysis (PCA). Following face detection, 26 key facial points (e.g., eye corners and mouth corners) are tracked using algorithms reported in [8]. The tracking results provide the location, size, and orientation estimates of the mouth. These parameters are subsequently smoothed over time and used to determine a 64×64 -pixel ROI.

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 through a pipeline consisting of intra-frame LDA/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 q to give the final visual feature vectors (VI stream) with 41 dimensions.

The basic audio features extracted by the front-end are 24-dimensional Mel-frequency cepstral coefficients. After cepstral mean normalization, nine consecutive frames are concatenated and projected onto a 60-dimensional space through an LDA/MLLT cascade, generating the AU feature stream.

The AVf features are generated by concatenating the 60-dimensional AU and the 41-dimensional VI features and projecting this 101-dimensional feature to a 60-dimensional subspace through LDA/MLLT [5].

The recognition system uses three-state, left-to-right phonetic HMMs with context-dependent states. The instances of the sub-phonetic states are identified by growing a decision tree that clusters left and right contexts spanning up to five phones on each side. The states are specified by the terminal nodes of the tree, and the corresponding observation streams are modeled by mixtures of Gaussian densities with diagonal covariance matrices.

5. Experimental Evaluations

5.1. Experimental Setup

The audio-visual speech recognition system is evaluated on a connected-digit recognition task using the IBM studio-DIGIT audio-visual database [5]. The corpus consists of full-face frontal video of 50 subjects, uttering 7 and 10-digit strings. A total of 6.7K utterances are recorded in a studio environment with uniform background and lighting. The acoustic signal to noise ratio (SNR) of the recorded data is measured at 19.5 dB.

The dataset is partitioned into three subsets: a training set containing 5.4K utterances, a test set with 623 utterances, and a held-out set including 663 utterances.

To evaluate the recognition performance in noisy environ-

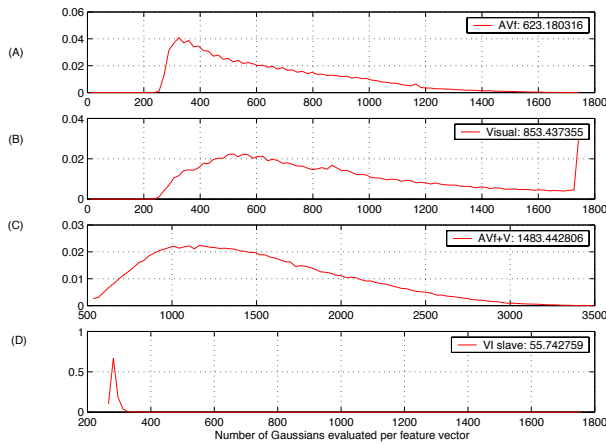


Figure 3: Gaussian usage for (A) audio-visual fused stream, (B) visual stream. (C) Combined usage for audio-visual fused (AVf) and visual streams operating independently. (D) AVf and visual under co-occurrence framework for audio SNR = 8.5dB.

ments, two noisy acoustic conditions were simulated by adding random segments of speech babble recordings to the clean speech samples. The average SNR of all three test conditions are 19.5dB (original), as well as, 11.5dB and 8.5dB (noisy). The HMMs are trained using the clean data, based on a context tree with 159 leaves modeled by 3.2K Gaussian densities.

5.2. Experimental Results

The baseline recognition accuracy of the three individual streams is shown as a function of SNR in the top three rows of table 1. The fourth and fifth rows of this table show results of the traditional independent multi-stream configurations for AU+VI and AVf+VI pairs.

To carry out the co-occurrence experiments, we generated two maps: one with AU stream1 and VI as stream2, and the other with AVf as stream1 and VI as stream2. These maps were generated from all of 5.4K training sentences. During run time, hierarchical labeling of stream1 was first carried out to generate the set \mathcal{Y}_1 . The set \mathcal{Q}_1 was then derived from \mathcal{Y}_1 by keeping only the Gaussians which attained the max score in equation 2. \mathcal{Q}_1 , in conjunction with the training time co-occurrence maps (Figure 1), is then used to identify the stream2 Gaussians that are to be evaluated (Figure 2), equation 8.

Figure 3 shows a normalized histogram of the number of Gaussians evaluated per feature vector of the AVf stream in panel (A) and for the VI stream in panel (B). We note that the sharp rise of the histogram at the trailing edge of (B) is due to absolute cutoffs on number of Gaussians that are permitted to be evaluated in the hierarchical labeler [7]. Panel (C) shows the histogram of the Gaussian usage for the case of independent combination of AVf and VI. The legend numbers in the plots indicate the mean usage per observation vector. As an aside, we note that in addition to measuring the computational load, these histograms also serve as an indicator of the Gaussian separability in different streams. For instance, a sharper distribution in panel (A) as compared to that in panel (B) indicates that the AVf stream has a better discrimination between Gaussians than the VI stream. This is in fact corroborated by the significantly lower error rate we obtain with the AVf system as shown in table 1.

Panel (D) of Figure 3 shows the Gaussian usage for the VI stream operating under the co-occurrence framework with AVf as stream1. The drastic reduction in Gaussian usage is evident from this figure. Details of the impact on Gaussian usage for the AVf and VI streams operating independently and with co-

| SNR | 19.5dB | 11.5dB | 8.5dB | Average |
|---------|--------|--------|-------|---------|
| AVf | 423 | 584 | 623 | 543 |
| VI Ind. | 853 | 853 | 853 | 853 |
| VI Co. | 30 | 48 | 56 | 45 |

Table 2: Average number of evaluated Gaussians per frame for the VI stream determined independently or as a slave of the AVf stream, compared with that of the AVf stream.

occurrence as a function of decreasing SNR is shown in table 2. On average we see a 94.7% reduction in the number of Gaussians evaluated while maintaining the word error rate of the independent stream result, as seen by comparing AVf+VI(Ind.) and AVf+VI(Co.) rows of table 1.

We note also from table 1 that for the case of AU+VI(Co.) there is a degradation as compared to the independent stream case at higher noise. This may be due to a weaker dependence between the AU and VI streams as compared to the AVf and VI pair; we are in the process of identifying the precise reason for this loss.

6. Conclusions

We presented a novel method for significantly reducing the number of Gaussian likelihood calculations in a multi-stream system through a method we call co-occurrence. On an audio-visual digit recognition task we find that for certain stream pairs large reduction in the number of Gaussian evaluations can be achieved without any loss in accuracy.

7. 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] Bourlard, H. and Dupont, S., “A new ASR approach based on independent processing and recombination of partial frequency bands,” *Proc. Int. Conf. Spoken Lang. Processing*, pp. 426–429, 1996.
- [3] Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., *The HTK Book*. United Kingdom: Entropic Ltd., 1999.
- [4] Dupont, S. and Luettin, J., “Audio-visual speech modeling for continuous speech recognition,” *IEEE Trans. Multimedia*, 2(3): 141–151, 2000.
- [5] 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.
- [6] Connell, J.H., Haas, N., Marcheret, E., Neti, C., Potamianos, G., and Velipasalar, S., “A real-time prototype for small-vocabulary audio-visual ASR,” *Proc. Int. Conf. Multimedia Expo*, pp. 469–472, 2003.
- [7] Novak, M., Gopinath, R.A., and Sedivy, J., “Efficient hierarchical labeler algorithm for Gaussian likelihoods computation in resource constrained speech recognition systems,” available on-line at: <http://www.research.ibm.com/people/r/rameshg/novak-icassp2002.ps>
- [8] Senior, A.W., “Face and feature finding for face recognition system,” in *Proc. Int. Conf. Audio Visual-based Biometric Person Authentication*, pp. 154–159, 1999.