Protecting the Mission: Hidden Semi-Markov Models for Visual Speech Recognition

Caner Berkay Antmen
berkay.antmen@mail.utoronto.ca

Eric Bannatyne
eric.bannatyne@mail.utoronto.ca

1 Introduction

Visual speech recognition (VSR) is the problem of recognizing the sequence of shapes produced by a speakers mouth in order to determine what words are being spoken. Humans have used lip reading for centuries as an aide for those with hearing disabilities, and visual input plays a major role in normal human speech perception [1]. The vast majority of research in the area of automatic speech recognition (ASR) has focused on traditional audio-based speech recognition. Although a number of researchers have attempted to use video to supplement sound in performing audio-visual speech recognition (AVSR), less work has been done to attack the problem of developing models that can automatically perform lip reading using video alone. A recent survey of methods used for VSR can be found in [2].

Lip reading is a very difficult task for humans and for machines alike, lending to its use as a plot device in popular comedies and science fiction movies [3]. The primary reason for such difficulty is the high degree of ambiguity that is inherent to recognizing speech from video alone, compared to audio-based speech recognition. This is because much of the variation in sounds that humans can produce is due to effects such as voicing, the position of the tongue and movement of the glottis, none of which are visible to an external observer. For example, a lip reader would have difficulty distinguishing between the words "peas" and "beets" when watching a speaker read a restaurant menu aloud. Human lip readers account for this problem by using contextual knowledge and knowledge of the language being spoken.

In our work, we apply hidden semi-Markov models (HSMM), a natural extension of regular hidden Markov models (HMM) accounting for random state durations, to the problem of visual speech recognition.

2 GRID Corpus

The GRID corpus is a freely-available audiovisual sentence corpus containing video, audio and textual transcriptions of sentences spoken by 34 speakers (16 female, 18 male), each uttering 1000 different sentences [4]. Each video is approximately three seconds long, consisting of 75 frames. In each video, the speaker's face is centered in the frame, facing towards the camera, on a solid background, as in Figure 1. We discarded a small number of videos that introduced noise into our data, for example a few videos contained compression artifacts that caused our video preprocessing system to fail. All sentences follow a regular syntactic structure and are of the form "Place blue by G7 now." Assuming that the sentences possess this type of structure enables our VSR system to focus more on mapping sequences of frames to words, rather than resolving ambiguity in the sentences. The textual transcriptions of the utterances include word-level alignments, allowing us to match words to the video frames in which they are being spoken, however the corpus does not include phoneme-level alignments.

Other audiovisual speech datasets include the AVLetters datasets, consisting of utterances of single letters of the English alphabet [5], and the Language Independent Lip Reading (LILiR) TwoTalk corpus, which includes four 12-minute long conversations between two speakers [6]. However, the

Figure 1: Speakers 1 and 4 in the GRID audiovisual sentence corpus.

AVLetters datasets are much smaller and simpler than the GRID corpus, whereas the LILiR TwoTalk corpus possesses a much richer structure and would thus require a much more involved analysis. The GRID corpus provides a good balance in terms of the structural complexity of the utterances, and the amount of data available for training and evaluation. Another motivation behind using GRID corpus is that there are no dependence between the words spoken and thus transition probabilities between words are uniform. Therefore a language model cannot be used to get an advantage in this corpus, which puts more emphasis on visual challenges it provides.

2.1 Data Preprocessing

Before training our HSMM visual speech models to recognize speech from video input, we first applied a series of preprocessing steps to the GRID corpus in an attempt to extract useful features for lip reading.

2.1.1 Face and Mouth Detection

In order to perform VSR, we only need to pay attention to the region of the video around the speaker's mouth, and we can safely disregard everything else. In order to achieve this, we use Haar feature-based cascade classifiers implemented in OpenCV,[1] which includes cascade classifiers for mouth and face detection, to find the speaker's mouth in every video frame. The Haar classifier for mouth detection outputs a collection of rectangles that have a high probability of being a mouth.

One of the issues that arises in the use of Haar classifiers for mouth detection is the need to select a single rectangle among several candidates for the location of the speaker's mouth. To remedy this issue, we employ a heuristic to select a good mouth candidate. First we use a face detection Haar classifier to fit a rectangle to the speaker's face. In practice, the results of face detection seem to be much more reliable than mouth detection, and to select the best face candidate we simply choose the candidate rectangle with the greatest area produced by the Haar face detector. Then, given a set of rectangle candidates for the location of the speaker's mouth, we first discard all candidates such that 30% or more of their area lies outside of the detected face rectangle. Out of the remaining rectangles, we select the one with the lowest (in terms of position on the image) vertical midpoint, since we know that the speaker's mouth will always be in the bottom half of the frame. Finally, for every frame of every video we extract a fixed-size square image (in our case, 50 by 50 pixels) centered at the centroid of the best mouth rectangle candidate. The results of this process can be seen in Figure 2.

2.1.2 HOG Processing

To obtain our features for VSR, we apply Histograms of Oriented Gradients (HOG) [7] to grayscale versions of the lip images extract from every video frame. HOG extracts features by dividing the lip detection into cells and measuring oriented gradients for each cell. The primary reason for using HOG as our feature descriptor is because they are easy to compute and they have been shown to

[1] `http://opencv.org/`

Figure 2: Face and mouth detection for speakers 1 and 4 in the GRID corpus.

be successful in a variety of classification tasks: [8] describes a number of experiments using HOG, among other feature descriptors such as SIFT, for performing VSR, and found HOG to be the better of the two feature descriptors. This motivates the use of HOG in our work.

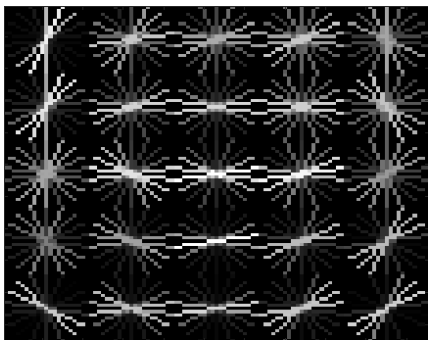

Figure 3: HOG features for the detected lips of speaker 4 in Figure 2.

We use the VLFeat library [9] to obtain our HOG features, which describe the lips detected during our image preprocessing stage. For our application, we used a HOG cell size of 10, which gives a tensor of dimensions $5 \times 5 \times 31$ for each frame. For implementation reasons, this tensor is flattened into a 775-dimensional vector in $\mathbb{R}^{775}$. Our cell size is motivated by the findings in [8], however we did not have the resources to run a detailed experiment for our purposes.

A potentially better alternative for computing feature descriptions may involve using Active Appearance Models (AAMs) on lips. However we were unable to apply this method due to a lack of lip landmark annotations.

3 Hidden Semi-Markov Models for VSR

In using a traditional hidden Markov model, one models sequential data as having an underlying sequence of unobserved states, each of which emits a single observation, with the assumption that the hidden state sequence is a Markov process. HMMs are frequently used in traditional audio-based automatic speech recognition applications, where the hidden states may correspond to the phonemes (or sub-phones), and the observations might, for example, consist of feature vectors containing mel-frequency cepstral coefficients, which can be obtained from the Fourier spectrum of an audio signal [10].

One consequence of the Markov assumption is that, whenever the Markov process is in a given state, the probability that it stays in the same state over several time steps decays exponentially over time, following a geometric distribution. In many applications though, this is not a realistic model in cases when one would not expect state durations to follow a geometric distribution. Hidden semi-Markov models address this issue by explicitly modelling state durations [11]. Instead of being

constrained to emit a single observation, every observation in an HSMM can emit a sequence of states, the length of which is sampled from a duration distribution depending on the hidden state. Typical choices for the duration distribution in HSMMs are the Poisson distribution and the negative binomial distribution [12]. In an HSMM, we make the additional simplifying assumption that, within a given segment, the individual observations are conditionally independent given the hidden state. More robust, but also more complex models such as segmental HMMs exist, which can model arbitrary joint distributions for the observations within a given segment [11]. However we shall focus our attention on the usual, albeit somewhat more restrictive, HSMM formulation.

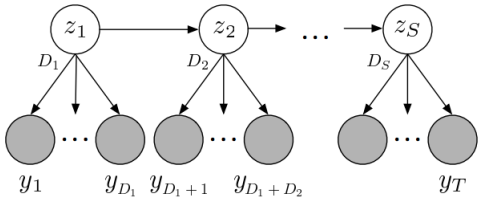

Figure 4: DGM of a HSMM with hidden states $z_1, \ldots, z_S$, durations $D_1, \ldots, D_S$, and observations $y_1, \ldots, y_T$. Note that the structure of the DGM is itself random, with state durations that depend on the hidden state. Image by Johnson and Willsky [12].

The directed graphical model in Figure 4 describes the joint distribution of the random variables in our HSMM, with hidden states $z_1, \ldots, z_S$, durations $D_1, \ldots, D_S$, and observations $y_1, \ldots, y_T$. Each state duration D_i can be viewed as a random variable depending only on the hidden state z_i. In our application of HSMMs to VSR, we model our visual speech data as a word-level HSMM; the hidden state z_i represents the ith word uttered by the speaker, D_i the number of frames corresponding to the word z_i, and y_j is the vector of HOG features for the jth video frame. Let V denote the size of the vocabulary. In the context of the GRID corpus,[2] $V = 54$.

In our model, the underlying hidden Markov process $Z = (z_1, \ldots, z_S)$ is parameterized by a vector of initial state probabilities $\boldsymbol{\phi} \in [0, 1]^V$ and a transition matrix $A_{V \times V}$. For each state z_i, the duration D_i of state z_i is modelled using a Poisson distribution $D_i \mid z_i \sim Poisson(\lambda_i)$. Finally, we model the observed HOG vectors using mixtures of Gaussians. Specifically, the observations associated with the hidden state z_i are modelled as a mixture of Gaussians with a fixed number of mixture components K, mixture weights π_{ik}, means μ_{ik} and covariance $\boldsymbol{\Sigma}_{ik}$. Thus if z_i is the parent of the observation node y_j in the DGM, then the distribution of y_j given z_i is

$$p(y_j \mid z_i) = \sum_{k=1}^{K} \pi_{ik} \mathcal{N}(y_j \mid \mu_{ik}, \boldsymbol{\Sigma}_{ik}).$$

The reasoning behind modelling the observations using mixtures of Gaussians is that our model should attempt to capture variations in the visemes produced by speakers (i.e. shape of a speaker's mouth when they produce a certain sound). Thus we expect the distribution of visemes corresponding to a given word to be multimodal. Although it is not explicitly denoted as such in the DGM in figure 4, we treat each state duration D_i as a random variable depending only on the state z_i. Moreover, although the underlying graph structure of the DGM is itself, strictly speaking, random (since the edges from hidden states to observations depend on the randomly distributed durations), for the purposes of parameter estimation we can treat it as being fixed, since the word-to-frame alignments in the training data are already known.

In order to train our model, we find values for our model parameters so as to maximize the likelihood of our data. Let $\theta = (\boldsymbol{\phi}, A, \lambda, \pi, \mu, \boldsymbol{\Sigma})$ be the vector of all of our model parameters. Based on the structure of the DGM for a HSMM, the joint likelihood factors as follows.

$$p(Y, Z, D \mid \theta) = p(z_1 \mid \boldsymbol{\phi}) \left(\prod_{i=2}^{S} p(z_i \mid z_{i-1}, A) \right) \left(\prod_{i=1}^{S} p(D_i \mid z_i, \lambda_i) \right) \left(\prod_{i=1}^{S} \prod_{j=t_i}^{t_i'} p(y_j \mid z_i, \pi_{ik}, \mu_{ik}, \boldsymbol{\Sigma}_{ik}) \right),$$

[2] We distinguish silence at the start of an utterance from silence at the end of an utterance in order to enforce the constraint that a sequence of video frames corresponds to exactly one sentence.

where $t_i = \sum_{\ell=1}^{i-1} D_\ell$ and $t'_i = \sum_{\ell=1}^{i} D_\ell$. Therefore we can maximize the likelihood by computing maximum likelihood estimates for each of the four factors individually. Estimating the initial state probabilities $\boldsymbol{\phi}$ is simple; in the GRID corpus every utterance begins with silence, with no exceptions. The transition matrix A can be computed almost as simply, from counts of bigrams that occur in the transcriptions of the training utterances. We can also compute maximum likelihood estimates for the λ_i parameters exactly for the duration distributions using the standard maximum likelihood estimator for the Poisson distribution:

$$\hat{\lambda_i} = \frac{1}{N} \sum_{\ell=1}^{N} D_i^{(\ell)},$$

where $D_i^{(\ell)}$ is the ℓth training example of the duration of the word $z_i^{(\ell)}$. To estimate the parameters π, μ and $\boldsymbol{\Sigma}$, we use expectation maximization to train a mixture of Gaussians for each word in the vocabulary.

There exist algorithms for performing inference in HSMMs based on blocked Gibbs sampling with message passing, introduced by Johnson and Willsky in [12, 13]. On a high level, the idea behind blocked Gibbs sampling involves partitioning the variables to be sampled into separate blocks, and then sampling from the joint distribution of each block, conditioning on all other variables [14]. In the context of a performing inference in an HSMM, this procedure involves sampling from the posterior state duration distributions. For a detailed description and analysis of such inference algorithms for HSMMs, see [12, 13].

3.1 HSMM Implementation

The implementation of our VSR system makes use of the Python library pyhsmm[3], which implements many of the algorithms found in [12]. It should be noted that most of the algorithms that pyhsmm implements are focused towards Bayesian inference in HSMMs, whereas our approach to VSR is mainly non-Bayesian in nature. However, at the time of writing, there are very few libraries for working with HSMMs apart from pyhsmm that are freely available. As such, we have modified several aspects of pyhsmm in order to suit the needs of our system.[4]

Instead of using pyhsmm to train our visual speech models, since the GRID corpus includes word-level text-to-video frame alignments, the parameters of the HSMM are trained separately via maximum likelihood estimation, described in Section 3, and a HSMM is initialized with those learned parameters. The Gaussian mixture models for the HOG video frame observations are trained using the implementation of EM in scikit-learn[5] [15].

Once the HSMM is trained, given a new sequence of video frames, we use the sampling algorithms implemented in pyhsmm to predict the sequence of words in the utterance. In order to make the Bayesian sampling algorithms implemented in pyhsmm work correctly for our non-Bayesian methods for VSR, we modified a number of components of pyhsmm to keep most of our model parameters fixed, rather than resampling them at every iteration of inference.

One of the important features of the GRID corpus is the rigid structure that the sentences possess. For example, every utterance begins and ends with silence, and every utterance contains exactly one sentence. In order to enforce these constraints in our model, we introduce a new word in our vocabulary to distinguish between silences occurring at the start of an utterance and silences occurring at the end of an utterance. If we do not introduce this distinction, then the model may predict multiple sentences within the same utterance, exhibiting cyclic behaviours in the predicted state sequences. However, to ensure that this modification does not harm the model's ability to recognize silences, the training data for frames of silence are combined into a single dataset, and a single Gaussian mixture model is trained to account for both kinds of silence.

A limitation of our implementation is the fact that it does not account for the length of a state sequence, whereas every utterance in the GRID corpus consists of exactly 8 words (including silences).

[3] `https://github.com/mattjj/pyhsmm`

[4] The implementation of our VSR system is freely available on GitHub at `https://github.com/aldld/lip-reading`.

[5] `http://scikit-learn.org/`

Instead, during prediction, our HSMM samples the sequences of states based on the observations and the duration distributions. Our implementation could be improved by incorporating fixed-length state sequences to more accurately model GRID corpus sentences.

4 Experiments

As mentioned before, the GRID corpus has 34 speakers with 1000 videos for each. For our purposes, we used medium quality (360x288) videos.[6] We discarded videos for which no face or lips were detected. In the end we had about 32550 videos remaining which amounts to around 970 videos per speaker.

We used an 80/20% training/testing split of the speakers, and thus our training set consisted of videos for 26 speakers, and our test set contained 7 speakers. We used our training set to train our model with different numbers K of mixture components, with all other parameters held fixed. We considered values of $K \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$. We measured the performances of our models on the test set by computing the Word Error Rate (WER) between the ground truth of words spoken per frame and our inference. Likewise, we computed the intersection-over-union between the ground truths and our inferences.

Furthermore, we picked two of the parameter settings that yielded better models and then trained those models only on differing subsets of training set to observe the effect that this would have on the models' the performance on the test set.

5 Results

Table 1 shows how our models performed for different values K for the number of Gaussian mixture components, with respect to the WER and IOU metrics.

K	WER	WER Std. Dev.	IOU	IOU Std. Dev.
2	0.6862	0.0925	0.2486	0.0948
4	0.6406	0.1001	0.2499	0.0989
6	0.6438	0.1076	0.2353	0.0916
8	0.8254	0.0876	0.1726	0.0893
10	0.8484	0.0701	0.1721	0.0908
12	0.6436	0.0957	0.2502	0.0982
14	0.8004	0.0843	0.2481	0.0945
16	0.8284	0.072	0.1622	0.0768
18	0.7938	0.0766	0.2502	0.0992
20	0.6486	0.1047	0.2485	0.0987

Table 1: Average word error rates and intersection-over-union (Jaccard index) metrics for various numbers of mixture components K.

Future work could calculate the expected IoU and WER measurements given from random sampling over 8 words, without replacement, with and without accounting for the sentence structure of GRID corpus. We did not have the resources to do such calculations but we believe that our results are significantly better than chance.

Table 2 shows the test set accuracy of our models trained on smaller subsets of the training set.

We observe that the accuracy diminishes when only 40% of the data is used for training for both 4 and 12 mixture components. We hypothesize that the reason for this is that some of the speakers that are introduced when the training set consists of more than 20% of the total data are handled incorrectly by our preprocessing system, either during lip detection or during HOG feature extraction. This hypothesis is further supported by the jump in the accuracy after we the training set consists of more than 60% of the data.

[6] Speaker 21's videos were missing from the dataset.

K	%Training data	WER	WER Std. Dev.	IOU	IOU Std. Dev.
4	20%	0.7775	0.0857	0.249	0.0955
4	40%	0.846	0.0737	0.1719	0.0925
4	60%	0.7731	0.0873	0.2487	0.0986
12	20%	0.747	0.0872	0.2483	0.0947
12	40%	0.8347	0.0754	0.249	0.0968
12	60%	0.697	0.0909	0.2499	0.0975

Table 2: Average WER and IOU for varying numbers of mixture components K and training set sizes.

Furthermore, the accuracy results in Table 1 are strictly better than those we observe in Table 2. This indicates that our model capacity is high enough to avoid overfitting and to generalize to unseen data.

6 Discussion and Conclusion

In our model we have not fully adopted the power of Bayesian parameter estimation[7] and inference for our HSMM-based VSR models. It would be interesting to experiment with Bayesian HSMMs and various extensions such as hierarchical Dirichlet process HSMMs for model selection and Bayesian inference, as outlined in [12].

Furthermore, it should be noted that forced-alignment could be used to align phonemes with the given video frames to investigate the properties of a phoneme-level HSMM as opposed to a word-level HSMM. However, this requires the use of an acoustic model and is therefore left as an area of future investigation.

Feature descriptors other than HOG could be investigated, for instance SIFT features, in order to find the optimal feature descriptor for the task. Moreover, experiments could be run to determine the efficacy of using different HOG cell sizes, as the cell sizes that we used were motivated by recent work, as opposed to our own experimentation.

In this work we used two error metrics: IoU and WER. However, it is difficult to find the best possible error metric for VSR, as there is no clear heuristic for determining the extent to which a model's predictions deviate from the ground truth. Future work could focus on combining multiple error metrics as well as qualitative evaluation to find useful metrics for evaluating VSR.

Lastly, from our experience working with existing HSMM codebases, we have observed a lack of useful and freely-available general-purpose HSMM libraries, which could be an interesting problem for future investigations into the theory and applications of HSMMs.

References

[1] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[2] Ahmad B. A. Hassanat. Visual speech recognition. *CoRR*, abs/1409.1411, 2014.

[3] Stanley Kubrick. 2001: A Space Odyssey, 1968. Screenplay by Stanley Kubrick and Arthur C. Clarke.

[4] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006.

[5] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, Feb 2002.

[6] Eng-Jon Ong, Richard Bowden, and Guildford GU27XH. Robust lip-tracking using rigid flocks of selected linear predictors. In *8th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2008.

[7] `https://github.com/JasperSnoek/spearmint`

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[8] Michael Wand, Jan Koutník, and Jürgen Schmidhuber. Lipreading with long short-term memory. *arXiv preprint arXiv:1601.08188*, 2016.

[9] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM, 2010.

[10] Mark Gales and Steve Young. The application of hidden markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304, 2008.

[11] Kevin P Murphy. Hidden semi-markov models (hsmms). *Technical report*, 2, 2002.

[12] Matthew J. Johnson and Alan S. Willsky. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14:673–701, February 2013.

[13] Matthew J Johnson and Alan Willsky. The hierarchical dirichlet process hidden semi-markov model. *arXiv preprint arXiv:1203.3485*, 2012.

[14] Deepak Venugopal and Vibhav Gogate. Dynamic blocking and collapsing for gibbs sampling. In *Uncertainty in Artificial Intelligence*, page 664. Citeseer, 2013.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.