

# Asymmetrically Boosted HMM for Speech Reading

Pei Yin, Irfan Essa, James M. Rehg

Georgia Institute of Technology  
GVU Center / College of Computing  
Atlanta, GA 30332-0280 USA  
{pyin, irfan, rehg}@cc.gatech.edu

## Abstract

*Speech reading, also known as lip reading, is aimed at extracting visual cues of lip and facial movements to aid in recognition of speech. The main hurdle for speech reading is that visual measurements of lip and facial motion lack information-rich features like the Mel frequency cepstral coefficients (MFCC), widely used in acoustic speech recognition. These MFCC are used with hidden Markov models (HMM) in most speech recognition systems at present. Speech reading could greatly benefit from automatic selection and formation of informative features from measurements in the visual domain. These new features can then be used with HMM to capture the dynamics of lip movement and eventual recognition of lip shapes. Towards this end, we use AdaBoost methods for automatic visual feature formation. Specifically, we design an asymmetric variant of AdaBoost M2 algorithm to deal with the ill-posed multi-class sample distribution inherent in our problem. Our experiments show that the boosted HMM approach outperforms conventional AdaBoost and HMM classifiers. Our primary contributions are in the design of (a) a boosted HMM and (b) asymmetric multi-class boosting.*

## 1. Introduction and Related Work

Speech reading<sup>1</sup> has been the subject of much research due to its obvious benefits in assisting speech interpretation by machines, especially when environmental noise is present in the audio channel [15]. A thorough review of speech reading can be found in [10]. The success of the hidden Markov model (HMM) [12] in speech recognition tasks has led to its application in computer vision domains. In particular, HMM has been used to analyze the lip motions in video imagery associated with speech. However, the performance of existing speech reading systems using HMM

<sup>1</sup>In this paper, we use the term speech reading for vision-based recognition of speech, much along the lines of lip reading, while speech recognition refers to conventional audio-based recognition.

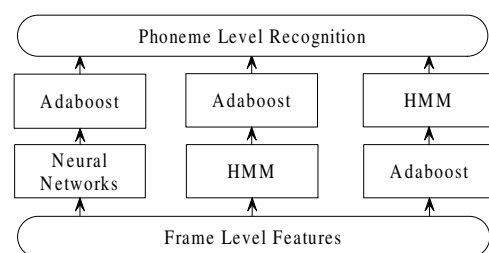


Figure 1: A simple top-level comparison with previous work on boosting in acoustic speech recognition. Left: Schwenck [14]. Middle: Meyer [8]. Right: Our system for speech reading.

is not at the same level as that of speech recognition, and mostly “address simple recognition tasks, such as small vocabulary ASR or isolated or connected words” [10]. One possible reason is that the visual features which can be used for recognition are insufficient to convey as much information as acoustic features. The alternative interpretation is that we have not found the appropriate features, analogous to Mel frequency cepstral coefficients (MFCC) in the audio domain, to model the video signal. In comparison to the audio domain, speech reading mostly relies on ad-hoc features. Therefore, the performance of such speech reading depends on visual features which are essentially not discriminative enough. In this paper, we present methods that allow automatic formation of informative features, which can then be used for recognition.

Recent work in face detection [17] has proposed a feature selection method based on AdaBoost. AdaBoost has also been used recently with HMM in the acoustic domain for speech recognition [14, 8] and with dynamic Bayesian networks in an audio-visual approach to speaker detection [5]. As shown in Figure 1, the previous work aimed at AdaBoost and HMM integration either relies entirely on AdaBoost for the phoneme classification while only using the HMM to form the higher-level language model [14] or simply con-

structs an HMM-AdaBoost ensemble by linearly combining the HMM [8]. While these approaches yield some good results, the first does not adequately model the inter-frame dynamics, and the second does not rely on statistically optimal features.

We propose an alternative integration called *boosted HMM*. It uses AdaBoost to automatically produce informative visual features and then feeds them to an HMM architecture, which in turn models the dynamics of lip movement. The role of AdaBoost in boosted HMM, is not only to adaptively concentrate on the hardest samples, but also to estimate the distribution of information among features with respect to classification for further integration. We also propose an asymmetric variant of AdaBoost M2 that can more effectively deal with the ill-posed multi-class sample distribution primarily due to the lack of discriminative visual features. To our knowledge, this is the first instance of integrating AdaBoost and HMM in such a manner, and the first attempt to address asymmetry directly in multi-class AdaBoost.

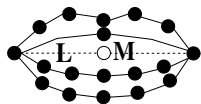


Figure 2: Definition of Reference Points(black)

## 2. Asymmetric Multi-class AdaBoost

The goal of speech reading is to process a sequence of video frames, and identify the corresponding *phoneme*<sup>2</sup> label among multiple candidates. As mentioned earlier, it is not quite clear what visual feature is effective for this classification problem, hence we need to extract some amount of raw features from the data set. As we need to eventually measure and model lip shape and motion, it is safe to assume that these features measure information about the lips. To this end, we have defined 18 reference points on the lip illustrated in Figure 2. Using the positions, velocities, accelerations, and the relative distance between those points, *etc.* we specify 168 raw features to describe one video frame. And we further assume that the unknown *discriminative* features lies in the space spanned by these measures. Inspired by [17], we associate a threshold to every raw feature to obtain a weak learner, so that AdaBoost can be applied to form discriminative features from the redundant *raw-feature* set<sup>3</sup>. Then the problem can be abstracted to multi-class classification with various raw-features.

<sup>2</sup>In our work, we segment the data stream based on phoneme boundary, sometimes referred to as a viseme [1].

<sup>3</sup>To prevent confusion, for the rest of the paper *raw-feature* refers to the raw visual features, and *feature* represents ones automatically formed by boosting.

### 2.1. Multi-class AdaBoost

AdaBoost [13] is the most widely used binary class boosting algorithm. It can generate a strong classifier (ensemble) with good generalization ability by linearly combining weak learners  $h^t$  selected from a sequence of optimization iterations  $t = 1, 2, \dots, T$ , weighted by their confidence. AdaBoost maintains a distribution  $D^t(i)$  over all samples and focuses specifically on the samples that are the hardest to discriminate [14]. It then finds the best weak learner  $h^t : X \rightarrow Y \in \{-1, 1\}$  from a predefined pool with respect to  $D^t(i)$ . Then according to the error made by the selected learner, its confidence is assigned, and  $D^{t+1}(i)$  are updated to concentrate on the more difficult samples.

To deal with the multi-class problem, AdaBoost M1 [4] is the direct extension from the binary case. At iteration  $t$ , it requires the weak learner to directly predict the class label  $h_{M1}^t : X \rightarrow Y \in \{1, 2, \dots, N\}$ . The ensemble then determines the class label for an instance with the maximum confidence summed across all iterations. Though simple to implement, it demands that there exist some weak learners with correct prediction greater than  $\frac{1}{2}$ . This is a much stronger requirement than “random” prediction, which only gives the accuracy of  $\frac{1}{N}$ .

An alternative approach, AdaBoost M2 [4] allows a set of weak learners<sup>4</sup> at iteration  $t$  to make a contribution to the ensemble even when their composite accuracy is just better than random, but not necessarily above  $\frac{1}{2}$ . Each weak learner addresses a specific class and outputs a likelihood value for every sample. A set of  $N$  weak learners provide the likelihood for all the sample-label pairs  $h_{M2}^t : X \times Y \rightarrow [0, 1]$ . And then the ensemble determines the class label of that instance by voting across all iterations. Thus the task for weak learner is just a *one-against-all* problem. The difference between AdaBoost M2 and simply constructing  $N$  separate one-against-all ensembles is that AdaBoost M2 holds the  $N$  binary problems inside its multi-class structure by sharing the same set of sample distribution; and provides a communication mechanism between the ensembles. This communication is achieved by introducing a set of *label weight functions* which represent how likely the given sample could be mistakenly assigned to every other class by the ensemble, and then minimizing a measurement called *pseudo-loss* which estimates how far behind the output of the correct class is in comparison to the “collaboration” of the wrong classes.

In other multi-class variants of AdaBoost, like AdaBoost ECC [7], multiple ensembles are built not corresponding to one class, but one bit of the error correcting code of the class prediction. These methods are designed for higher process-

<sup>4</sup>In [4], they are considered as one weak learner composed by a set of questions. We abuse the notation a little bit to make the discussion clearer.

ing efficiency by decomposing the problem into the binary case while achieving the similar level of accuracy as AdaBoost M2.

## 2.2. Limitations of AdaBoost M2 for Feature Selection

A logical way to apply AdaBoost for feature selection in speech reading would be to use AdaBoost M2 in conjunction with weak learners based on lip features. Although AdaBoost M2 introduces *pseudo-loss* to allow more subtle optimization, we have observed that it is still hard to extract a useful weak learner for the ensemble when the weak learners and associated raw-features have quite limited discrimination power. In such a case, the boosting procedure will “converge” quickly at rather low accuracy. In the application of speech reading with our raw-feature set, the multi-class classification problem is very hard. Figure 3 shows two smoothed histograms of the  $y$  position of the one control point on the upper lip. The lower black curve is the histogram of the samples for phoneme /AA/ enlarged by a factor of 10, while higher gray curve is histogram of the samples from all the other 38 phoneme classes. This example illustrates the general difficulty of “finding a needle in the haystack” in the classification problem. For every class, its samples distribute across most of the bins, while it dominates none of those bins.

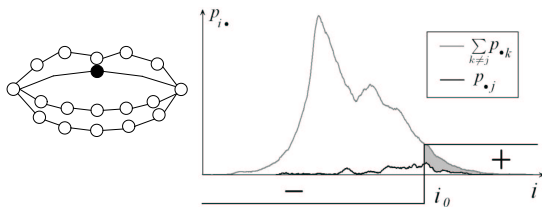


Figure 3: Left: the selected RP of the Lip Model; Right: the histogram of the  $y$  position of that RP.

To simplify the discussion, we assume that the weak learner will assign a threshold to separate positive class from the others. For AdaBoost M1, the optimization fails in the first round, because at every iteration, the weak learner is required to output a single prediction of class label for a specific bin. However there are no such weak learners that can achieve an accuracy greater than  $\frac{1}{2}$  at any threshold. No matter which class label is assigned, there are always more samples from other classes. In principle, AdaBoost M2 can address the decomposed binary classification problem. However, when the features are not sufficiently discriminative, the optimization approach of training the weak learner to maximize classification accuracy results in poor feature selection. As Figure 3 illustrates, assume the weak learner searches for an optimal threshold  $i_0$ , where  $p_{uv}$  is

the sample density of class  $v$  at bin  $u$ , and  $j$  is the current positive class, the optimization will be

$$\max_{i_0} \left\{ \sum_{i=1}^{i_0} \sum_{k \neq j} p_{ik} + \sum_{i=i_0}^n p_{ij} \right\} \quad (1)$$

But since  $N > 2$  in multi-class,  $p_{ij}$  (black curve) will be much lower than  $\sum_{k \neq j} p_{ik}$  (gray curve), then the net benefit

of increasing the threshold will be qualitatively<sup>5</sup> the shaded area, which results in  $i_0 = n$  *i.e.*, the weak learner with the highest accuracy is to set the threshold at the rightmost position in the figure, which classifies everything as negative.

The above example shows that AdaBoost M2 aggressively exploits only a few “easy” raw-features and gets stuck in local optimum quickly. Because of the competition of the ensembles thereafter, guessing the sample belongs to some class with slightly low confidence is more “informative” than guessing it is not for multi-class problem, although the second choice is safer, and leads to higher binary classification accuracy in this hard situation. Therefore, the risk of false positive and false negative is not weighted symmetrically in the multi-class classification, which is similar to the task of rare event detection, such as [18]. Unfortunately, this internal asymmetry is not taken into account in classic AdaBoost M2.

## 2.3. Asymmetric AdaBoost M2

To fully explore the information among the raw-features instead of to exploit a few, we introduce the asymmetric cost in multi-class AdaBoost. In [7], asymmetric AdaBoost ECC is presented, where asymmetric cost is applied in the decomposed binary classification, but we want to build asymmetry inside the multi-class boosting structure. In the asymmetric AdaBoost of [16], their Asymmetric Loss (*ALoss*) is to change the density of the positive/negative class, but this can not be directly applied in multi-class case. As we mentioned earlier, in contrast to  $N$  isolated one-against-all binary boosting, which has  $N$ -fold distribution of training samples, AdaBoost M2 only maintains 1-fold of distribution. Thus the trick of density modification in binary case does not work, because every class will be considered as positive class once, and the remaining is considered as negative class. If the density of one class is increased, the density of all the other classes have to be increased too. Then finally every class gets the same bonus, which is exactly the same as the symmetric AdaBoost M2.

We consider an alternative algorithm, where AdaBoost M2 retains its own density updating procedure, and only passes the asymmetrically modified “fake” distribution to

<sup>5</sup>Because the black curve is enlarged by 10 times.

the underlying weak learners to bias them towards the current desired classes, as shown below.  $A_t(i, y)$  is the asymmetric density factor of the  $i$ th sample to be assigned to class  $y$  at time  $t$ . It is greater than 1 when  $y = y_i$  to amplify the density of the positive class, and equal to 1 otherwise.  $U_t(i, y)$  is the updating factor for  $A_t(i, y)$ .

---

### Asymmetric AdaBoost M2

Input :  $\langle x_i, y_i \rangle$  with class label  $y_i \in Y = \{1, \dots, k\}$   
and sample distribution  $D_i$

Loop  $t = 1, 2, \dots, T$

$$W_i^t = \sum_{y \neq y_i} w_{i,y}^t; \quad q_t(i, y) = \frac{w_{i,y}^t}{W_i^t} \text{ for } y \neq y_i;$$

$$\text{and } D_t(i) = \frac{W_i^t}{\sum_{i=1}^N W_i^t}$$

Call WeakLearn, provide the distribution  $D_t^*(i, y) = D_t(i) \cdot A_t(i, y)$ , and label weighting function  $q_t(i, \cdot)$ ; get back the hypothesis  $h_t : X \times Y \rightarrow [0, 1]$

Compute the *pseudo-loss* of  $h_t$  :

$$\varepsilon_t = \frac{1}{2} \sum_{i=1}^N D_t(i) \left( 1 - h_t(x_i, y_i) + \sum_{y \neq y_i} q_t(i, y) h_t(x_i, y) \right)$$

Set  $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$

Update  $w_{i,y}^{t+1} = w_{i,y}^t \beta_t^{(1/2)(1+h_t(x_i, y_i)-h_t(x_i, y))}$ ,  
and  $A_{t+1}(i, y) = A_t(i, y) \cdot U_t(i, y)$

Output  $\arg \max_{y \in Y} \sum_{t=1}^T \left( \log \frac{1}{\beta_t} \right) h_t(x, y)$

---

To validate this approach, we first test it on synthesized data. The data corpus contains five classes, and every sample is represented by a feature vector with 20 elements, while every element can take one out of five values. For every element, the samples from the same class will always have the odds of 35%-45% to take one favored value of the five, and have uniform chance of taking the other four values. This value preference is selected randomly, but mutually exclusively for the five classes in every element. And the preference is independent over the 20 elements. Then this distribution has the same property as the speech reading data described above. We sampled 10000 instances from this distribution for training and another 10000 for testing. We believe that the experiment result <sup>6</sup> in Figure 4 shows that the our asymmetric variant of AdaBoost M2 is able to extract more information from the whole feature set.

We also evaluate our algorithm on real data with 39 classes. The task is to determine the phoneme label for every frame in our speech data set by its visual raw features, and the asymmetry is chosen as  $\frac{39-1}{1} = 38$ , so it generally

<sup>6</sup>Because the performance for training and testing are quite similar in all the experiments performed, we only include the testing curve hereafter.

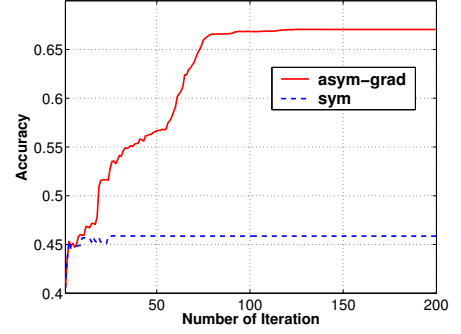


Figure 4: Tests on synthesized data show that asymmetric AdaBoost M2 performs better than the symmetric one

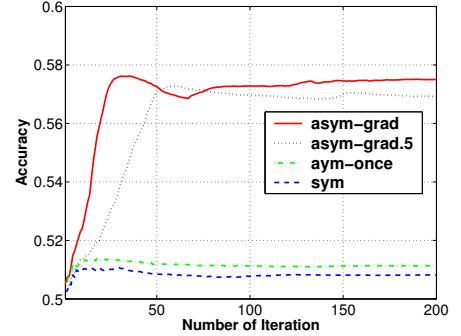
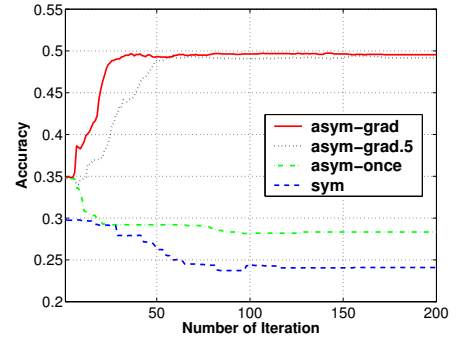


Figure 5: Tests on video (top) and motion capture (bottom) data show that asymmetric AdaBoost M2 performs better than the symmetric one

balances the overall density of the positive and that of negative samples for every class. The experiment result on video data and motion capture data are shown in Figure 5 (The algorithm process for this data is explained in Section 4). The results on real data again shows that asymmetric weak learner provides more information to boost.

We also performed the experiment applying the asymmetry all at once at the initial stage ( $U_t(i, y) = 1$ , “asym-once”) versus gradually increasing the density factor ( $U_t(i, y) > 1$ , “asym-grad”). And as shown in Figure 5, the second option yields better performance, consistent with [16]. And “asym-grad.5” represents the re-

sults when increasing the asymmetry factor at half speed of “asym-grad”.

Another choice to address asymmetry is to bias the classification threshold. But this method fails to encode the asymmetry property into the training procedure, and only makes the tradeoff *afterwards*. Experiments have shown that the indirect asymmetric method is not optimal [16].

### 3. Boosted HMM

As shown in detail later in Table 2 and Table 3, if the phoneme recognition is trivially performed by multiplying the frame-level likelihood from boosting output (AdaBoost only), the accuracy is low. The reason is that despite the existence of raw-features about short-term dynamics, boosting still can hardly capture the long-term dynamics. Therefore, HMM is then used to exploit this information.

To integrate HMM with AdaBoost, we first review the property of AdaBoost, and discuss what is indicated from the boosting outputs. Considering that the cues for classification are not uniformly distributed over all kinds of possible vision features, we rely on boosting algorithm to estimate such information about features, given the training samples. From the perspective of samples, AdaBoost tends to concentrate more and more on the samples that are hard to classify, and the reason is that those samples can provide more information about the decision boundary, in order to optimize the ensemble. And from the perspective of weak learners, AdaBoost assigns more weight to the learner that has higher accuracy. Recalling that in our case, for every class  $y$ , one weak learner is attached with one raw-feature, and the weak learners are from the same family, so that they have the same level of classification capacity. Hence the reason for AdaBoost to prefer one weak learner rather than another is that the associated raw-feature is more informative for classification; in other words, easier to classify. Combining the above interpretation, AdaBoost adaptively searches for the easiest feature for the hardest samples.

This suggests that AdaBoost addresses the non-uniform distribution of information not only qualitatively by selecting the weak learners over the informative features, but also quantitatively by the confidence assigned to the weak learners. Therefore, confidence of the weak learners, if normalized, can be considered as the distribution of the corresponding classification cues. Then such weighted raw-features are the most informative raw-feature set, which can be used to integrate with HMM. Unfortunately, HMM cannot take this kind of weighted raw-feature vector as input, since the components of which are not equally weighted.

One solution is to resample the raw-feature vectors, however Quinlan [11] claims that the resampling does not work as well as the reweighting. Alternatively, we convert the

weighted most discriminant raw-features via their corresponding classifiers to get the uniformly weighted, most discriminant outputs, as the features for the HMM. Actually, they are just in proportional to the outputs of the multi-class ensemble by a constant value.

If AdaBoost can always extract the correct class label, then using the confidence outputs of AdaBoost as features would be a trivial integration. However, it is hard to expect that to be true in the applications with dynamics, like speech reading. Our experiments show that the boosting method does not achieve very high accuracy at frame level, and performs much worse at phoneme level if not considering dynamic information, so taking results of other classifier as features for the HMM is a further exploitation of the data.

There are 39 phonemes in our data set, so a 39 dimensional feature space is computed for every video frame, then PCA is applied to further reduce the dimensionality and noise. Finally the features are assigned to the video frames to train one 3-state HMM with the mixture of Gaussian output per class.

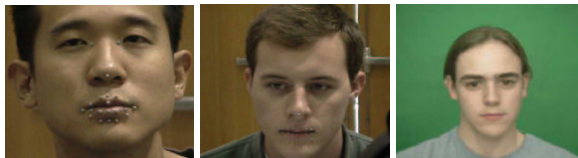


Figure 6: Three of the subjects used in our experiments. Left/Middle are subjects with motion capture markers, which allowed for high speed motion data (120Hz). Right is done using video (30Hz).

## 4. Experiments

### 4.1. Experimental Setup and Data

Our current experiments were performed on both video data and motion capture (MoCap) data with several subjects. We describe these data sets briefly:

**Motion Capture Data of Lips:** We used a Vicon Motion Capture System to capture over 30 minutes continuous speech motions from several subjects. The subjects were requested to wear markers on their lips along the same line as our reference points (RPs) in Figure 2. The RPs are tracked by infra-red tracker as shown in Figure 6. The 3D tracking results are then projected into the 2D plane determined by the extra trackers on the nose and beside the eyes.

**Visual Tracking of Lips:** We videotaped one subject talking continuously for almost seven minutes. The 2D position RPs were extracted using Eigen Points [2] approach.

As in Figure 2, the RPs are then aligned by making the line segment  $L$  between the two lip corners horizontal and its midpoint  $M$  at  $(0,0)$ . We then compute the extra parameters like the distance between two adjacent RPs, the center of the upper and lower lip, *etc.* Furthermore, the height, width, area covered, and roundness of the mouth, suggested by [6], are also included in the raw-feature set. Then we use Catmull-Rom splines [3] to smoothly interpolate the temporal trajectory of those parameters, and compute the velocity and acceleration from the splines.

We have thirty-seven minutes of speech data in total. The video data and the MoCap data are processed separately. Both these data sets provided us with a 168 dimensional raw-feature space for one visual frame. The experimental configuration is shown in Table 1.

**Audio-based Segmentation:** We first use CMU Sphinx<sup>7</sup> to segment the video into phonemes by forced alignment. Then for every phoneme with known boundary, the likelihood of the phoneme labels are estimated by the boosted HMM.

## 4.2. Results

We evaluate the classification performance by Cumulative Match Characteristics (CMC) [9], which describes “is the correct answer in the top  $n$  matches?” Experiments show that both symmetrically boosted HMM and asymmetrically boosted HMM effectively improve the accuracy of HMM and it also outperforms the method using AdaBoost only, as shown in Figure 7. The comparison between asymmetrically boosted HMM and other alternatives at the rank 1 is shown in Table 2 and Table 3.

Corpus	Total Len.	Training Data	Testing Data	Total Sent.	Total Phones
Video	6'30"	5'	1'30"	95	2322
MoCap	30'45"	24'42"	6'03"	275	8468

Table 1: Experiment Configuration.

Video Analysis	A. Boosted HMM	S. Boosted HMM	HMM Only	AdaBoost Only
Rank 1	39.38%	31.40%	20.33%	20.15%

Table 2: Rank 1 Comparison on Video Data

Video Analysis	A. Boosted HMM	S. Boosted HMM	HMM Only	AdaBoost Only
Rank 1	57.44%	45.16%	39.82%	20.37%

Table 3: Rank 1 Comparison on MoCap Data

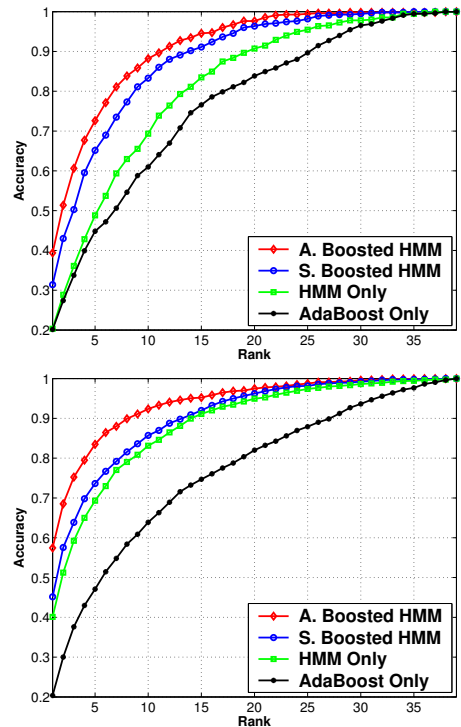


Figure 7: Comparison of asymmetrically boosted HMM, symmetrically boosted HMM, HMM only and AdaBoost only method on video data (top) and MoCap data (bottom).

In Table 2 and Table 3, “HMM Only” is HMM built on the 18 RPs positions, while “AdaBoost Only” is phoneme level classifier by simply multiplying the boosting likelihood of all the frames, “S. Boosted HMM” is the proposed novel integration of AdaBoost M2 and HMM, and “A. Boosted HMM” is the further improvement by introducing our asymmetric variant of AdaBoost M2.

By slightly changing the HMM parameters (number of Gaussians, number of states), we experimentally show that the success of the algorithm is not due to a magic parameter. Moreover, we have experimented with random raw-feature combination shown in Figure 8. The randomly formed features perform much worse than the best feature selected by AdaBoost. The results indicate that boosted HMM successfully takes advantage of both methods.

An undesirable issue of video data is the low video frame rate. 33% phonemes are so short that only one or even less than one frame is associated with them. In such a case, it is impossible for the HMM to get any dynamic cues from only one frame. To make things worse, it may be quite noisy and distorted by the adjacent phonemes (the coarticulation effect). The boosted HMM works 10% better if only phonemes with more than 2 frames are considered.

<sup>7</sup>Version 2-0.4 <http://fife.speech.cs.cmu.edu/sphinx>

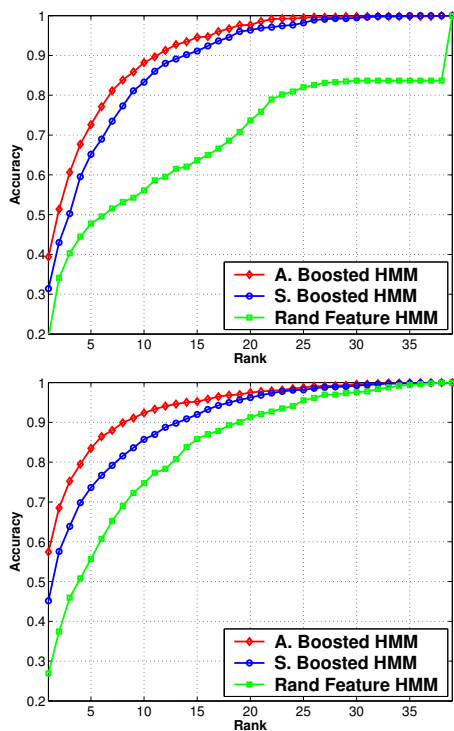


Figure 8: Comparison of HMM based on boosted features and random features on video data (top) and MoCap data (bottom)

## 5. Conclusion and Future Work

HMM is good at modeling the dynamics while AdaBoost is good at automatic feature selection. In this paper, we present a novel integration scheme, boosted HMM, for such a coupling of HMM and boosting. We also propose an asymmetric variant of AdaBoost M2, which deal with the ill-posed multi-class sample distribution better. Experiments show that the boosted HMM, especially asymmetrically boosted HMM outperforms both pure AdaBoost or HMM when used separately.

One shortcoming of our system is that the phonemes are pre-segmented by forced alignment by Sphinx system. So the HMM only needs to output phoneme likelihoods for a frame sequence with known boundaries. However, note that Sphinx is based on HMM too. So given more data to build the language model, boosted HMM can automatically do the segmentation themselves.

The current visual raw features can be considered as “semantic features”. In the future, we plan to integrate appearance features and other statistic features into the boosting procedure, design more powerful learners, and investigate the robust fusion with acoustic speech recognition.

## Acknowledgments

We would like to thank Matthew Mullin, Jianxin Wu and Howard Zhou for the helpful discussions and our subjects for volunteering their time. This work is in part funded by the NSF ITR grant #IIS-0205507.

## References

- [1] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *ACM SIGGRAPH'97*, pages 353–360, 1997.
- [2] M. Covell and C. Bregler. Eigen-points. In *Proceedings of 3rd IEEE ICIP*, volume 3, pages 471–474, 1996.
- [3] J. Foley, V. Dam, S. Fiener, and J. Hughes. *Computer Graphics, Principles and Practice*. Addison-Wesley, second edition, 1996.
- [4] Y. Freund and R.E. Schapire. A decision theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Science*, 55(1):119–139, 1995.
- [5] A. Garg, V. Pavlovic, and J. M. Rehg. Boosted learning in dynamic bayesian networks for multimodal speaker detection. In *Proceedings of the IEEE*, pages 1355–1369, September, 2003.
- [6] A. J. Goldschen, O. N. Garcia, and E. D. Petajan. Rationale for phoneme-viseme mapping and feature selection in visual speech recognition. In Stork and Hennecke, editors, *Speechreading by Humans and Machines: Models, Systems and Applications*. NATO/Springer-Verlag, New York, 1996.
- [7] V. Guruswami and A. Sahai. Multiclass learning, boosting, and error-correcting codes. In *the Twelfth Annual Conference on Computational Learning Theory*, pages 145–155. ACM Press, 1999.
- [8] C. Meyer. Utterance-level boosting of HMM speech recognizers. In *ICASSP*, 2002.
- [9] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), 1999.
- [10] G. Potamianos, C. Neti, G. Gravier, and A. Garg. Automatic recognition of audio-visual speech: Recent progress and challenges. *Proceedings of the IEEE*, 91(9), 2003.
- [11] J. R. Quinlan. Bagging, boosting, and C4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730. AAAI Press, 1996.
- [12] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, Prentice Hall, 1993.
- [13] R. E. Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.
- [14] H. Schwenk. Using boosting to improve a hybrid HMM/neural network speech recognizer. In *ICASSP99*, pages II:1009–1012, 1999.
- [15] D. G. Stork and M. E. Hennecke. *Speechreading by Humans and Machines: Models, Systems and Applications*. NATO/Springer-Verlag, New York, 1996.
- [16] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. In *NIPS*, 2001.
- [17] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, to appear, 2003.
- [18] J. Wu, J. M. Rehg, and M. D. Mullin. Learning a rare event detection cascade by direct feature selection. In *NIPS*, 2003.