

**Recognition of Visual Speech Elements
Using
Adaptively Boosted Hidden Markov Models**

·
Say Wei Foo, Yong Lian, Liang Dong

·
**IEEE Transactions on Circuits and Systems for Video Technology,
May 2004.**

Shankar Shivappa
University of California, San Diego
April 26, 2005

Overview

Automatic speech recognition (ASR) is a machine learning problem.

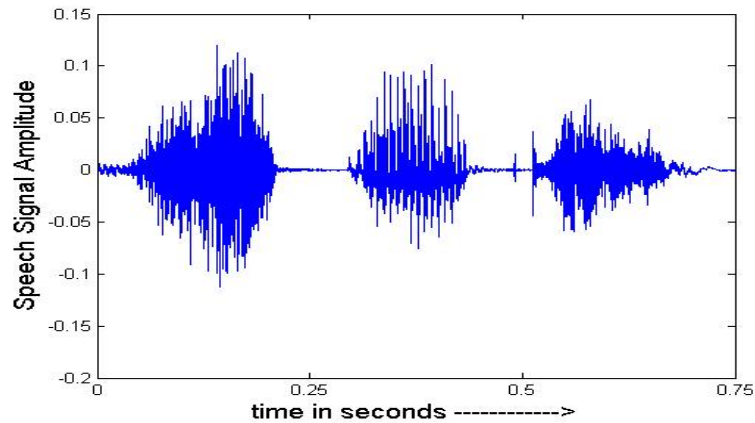
Hidden Markov models are widely used in ASR.

Visual speech (lip reading) can be used to improve robustness of speech recognizers.

Hidden Markov Models can be used to classify visual speech.

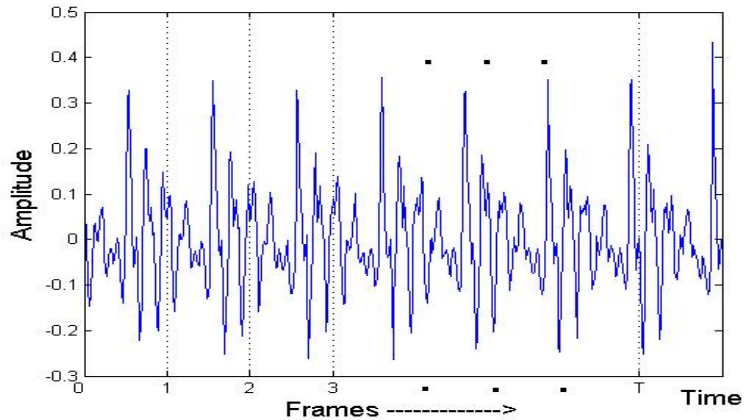
Boosting can be used with HMMs to improve performance.

Speech Recognition



Speech Recognition

The Speech signal is broken down into a sequence of frames, typically 30ms in duration.



Speech Recognition

For each 30ms frame, the feature extractor computes a vector of features called an “observation”.



Each word is a sequence of distinct sounds (phonemes).

Each phoneme is a sequence of feature vectors (observations).

The sequence of observations is recognized as one phoneme in the vocabulary by the pattern recognizer.

Visual Speech Recognition

Audio speech signals can be corrupted by background noise.

Speech production and perception is bimodal - audio and visual [3].

Lip reading uses the visual information to recognize spoken words.

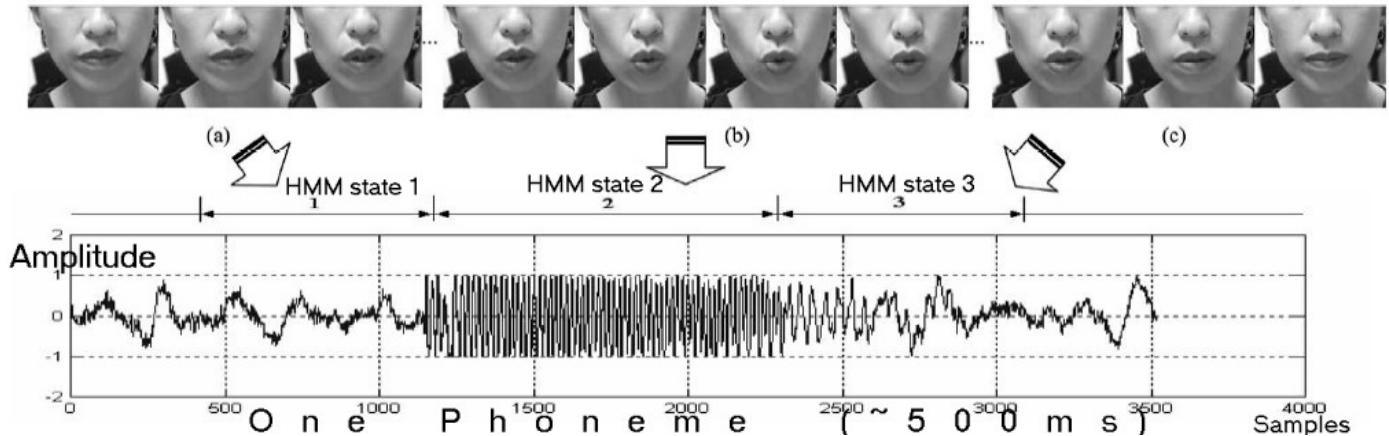
Visual Speech Recognition

The movement of mouth is captured in a series of video frames.

For each frame, relevant features are extracted to get a sequence of visual observations.

The basic unit of speech corresponding to a phoneme is a “viseme”.

The sequence of observations is recognized as a viseme in the vocabulary.

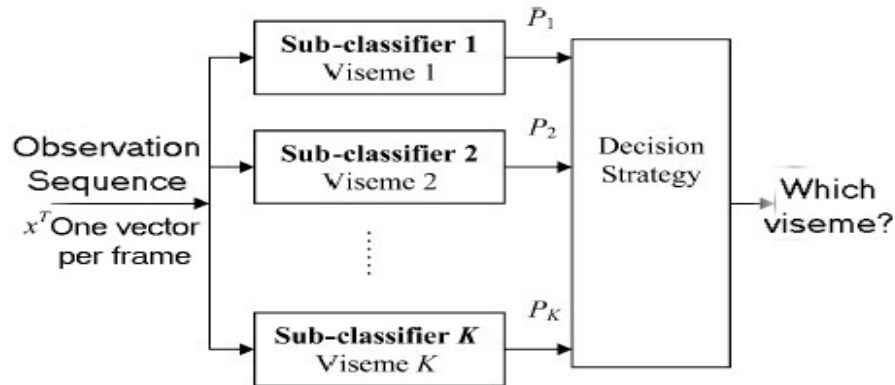


Visemes

WISEMES DEFINED IN MPEG-4 MULTIMEDIA STANDARDS

Viseme Number	Phonemes	Examples
0	none	(silence and relax)
1	p, b, m	<u>p</u> ush, <u>b</u> ike, <u>m</u> ilk
2	f, v	<u>f</u> ind, <u>v</u> oice
3	T, D	<u>t</u> hink, <u>t</u> hat
4	t, d	<u>t</u> each, <u>d</u> og
5	k, g	call, <u>g</u> uess
6	tS, dZ, S	<u>ch</u> eck, <u>jo</u> in, <u>sh</u> rine
7	s, z	<u>s</u> et, <u>z</u> eal
8	n, l	<u>n</u> ote, <u>l</u> ose
9	r	<u>r</u> ead
10	A:	<u>j</u> ar
11	e	be <u>d</u>
12	I	ti <u>p</u>
13	Q	sh <u>o</u> ck
14	U	g <u>oo</u> d

Multi-class Classification Problem

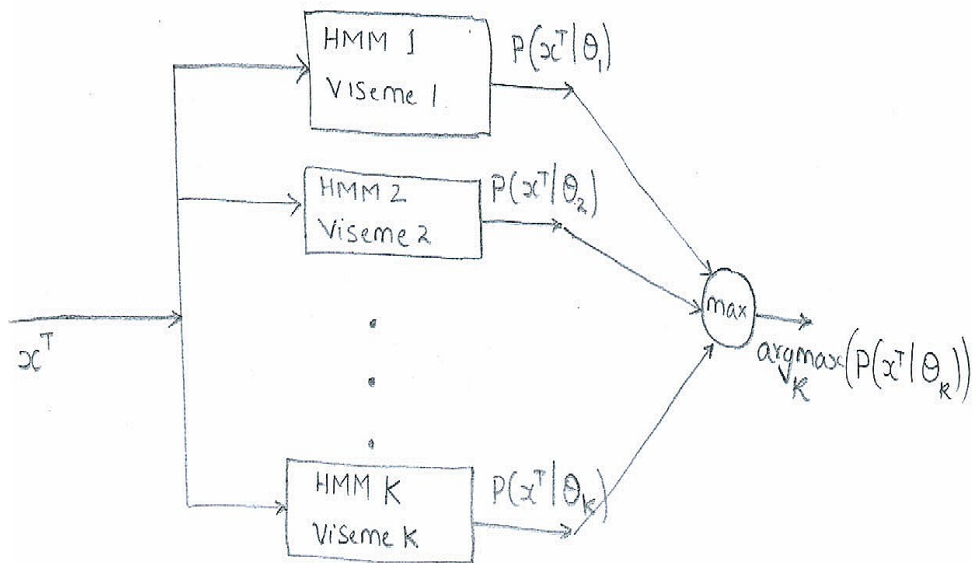


x^T means the observation sequence $\vec{x}^{(1)} \vec{x}^{(2)} \dots \vec{x}^{(T)}$

$$\begin{aligned} \text{Recognized viseme} &= \arg \max_k \left\{ P(\text{viseme}_k | x^T) \right\} \\ &= \arg \max_k \left\{ \frac{P(x^T | \text{viseme}_k) P(\text{viseme}_k)}{P(x^T)} \right\} \\ &= \arg \max_k \left\{ P(x^T | \text{viseme}_k) \right\} \end{aligned}$$

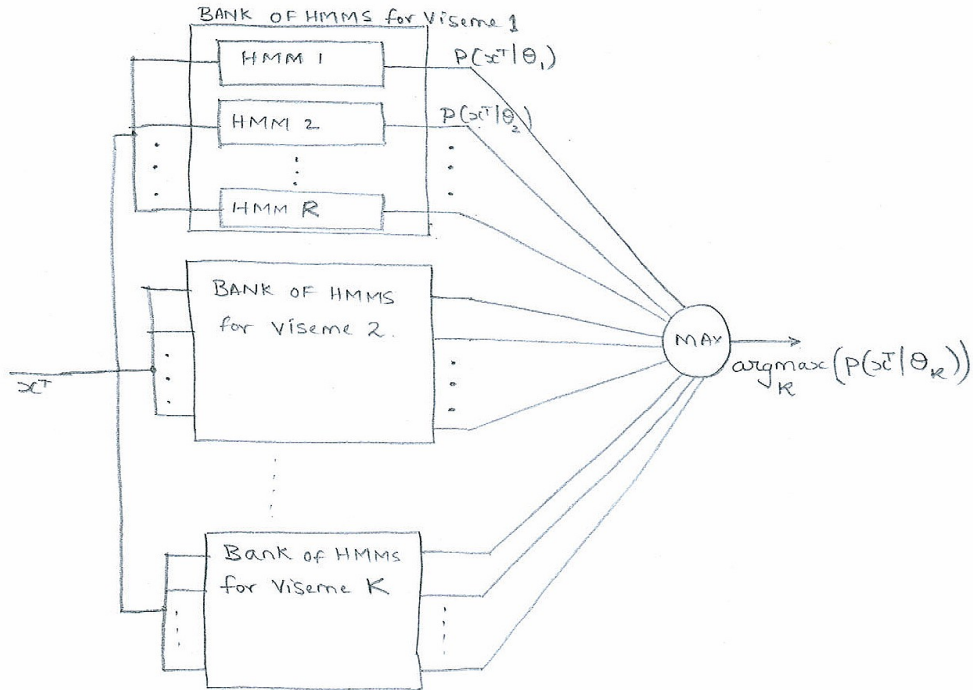
Multiclass Classification Problem

The model of $viseme_k$ is a HMM with parameter θ_k .



Multiclass Boosting

Actually, after t rounds of boosting, the model of $viseme_k$ is a bank of HMMs.



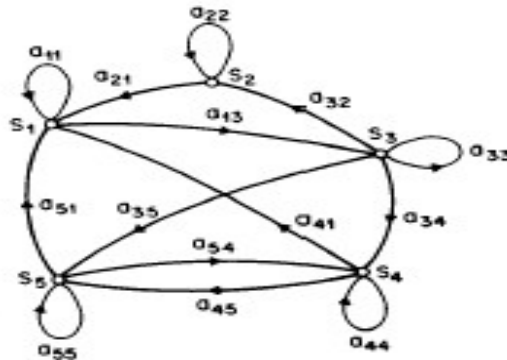
Markov Chains

The sequence of observations can be modeled by a hidden Markov model (HMM).

HMMs are extended finite state machines (automata).

At each timestep the system transitions from one state to another.

An example of a first order Markov chain:



The future state transition is independent of the past history and depends only on the present state.

Hidden Markov Models

The Markov chain is specified by an initial state distribution,

$$\pi = [\pi_i]_{1 \times N} = [P(s_1 = S_i)]_{1 \times N}$$

and the state transition matrix,

$$A = [a_{ij}]_{N \times N} = [P(s_{t+1} = S_j | s_t = S_i)]_{N \times N}$$

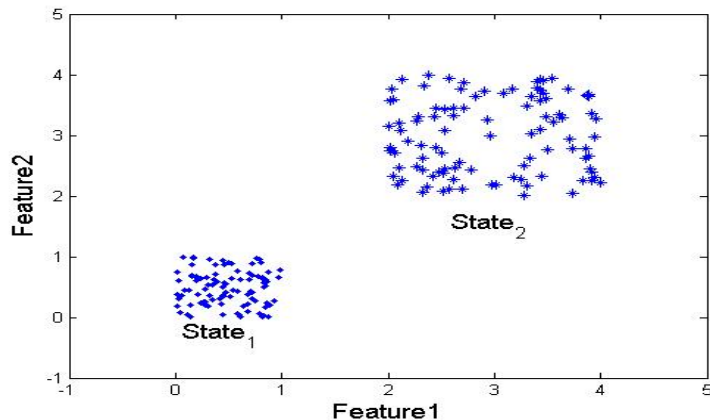
An HMM is a Markov chain where an observation is generated based on its current state, at each time step.

All we can see are the observations, not the states themselves.

Hence the name Hidden Markov model.

Hidden Markov models

The observations corresponding to a particular state are not the same, but are similar.



With M alternative observations at each state, the model specifies the observation probability matrix,

$$B = [b_{ij}]_{N \times M} = [P(O_j | S_i)]_{N \times M}$$

The HMM is specified by the parameters $\theta = (\pi, A, B)$.

HMM Classifier

Coming back to the HMM classifier,

$$\begin{aligned} \text{Recognized viseme} &= \arg \max_k \left\{ P(\text{viseme}_k | x^T) \right\} \\ &= \arg \max_k \left\{ P(x^T | \text{viseme}_k) \right\} \end{aligned}$$

Given the HMM θ_k for viseme k , we need to find $P(x^T | \theta_k) = P(x^T | \text{viseme}_k)$.

Note that the state sequence is unknown.

Therefore we need to sum over all possible state sequences Q ,

$$P(x^T | \theta_k) = \sum_{\text{all } Q} P(x^T | Q, \theta_k) P(Q | \theta_k)$$

There are N^T possible state sequences. Dynamic programming is used to compute $\sum_{\text{all } Q}$ efficiently [2].

Training One HMM

Training refers to learning the model θ_k from a training set of observation sequences

$$M = \{x_1, x_2 \dots x_R\}$$

The usual maximum-likelihood estimate is

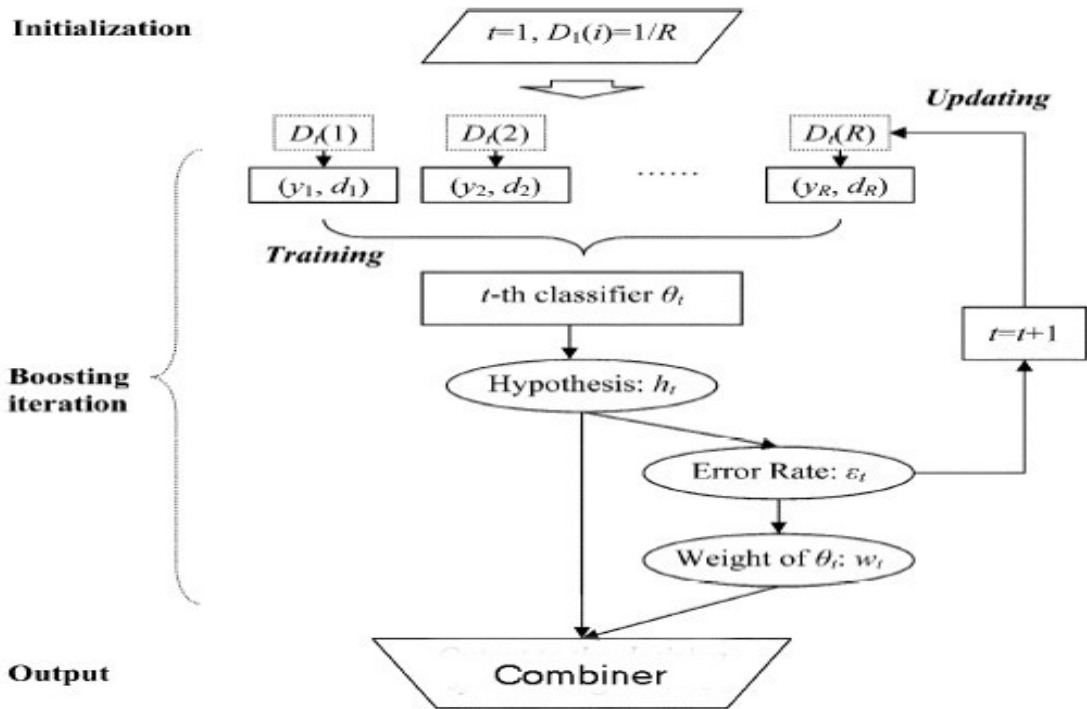
$$\hat{\theta} = \arg \max_{\theta} \left\{ \prod_{1 \leq i \leq R} P(x_i | \theta) \right\}$$

EM algorithm and dynamic programming give an iterative solution [2].

This process is called the Baum-Welch recursion.

Note that negative examples are not used to train the HMMs. This is true of ML estimation in general.

AdaBoost Review



Boosting HMMs

Boosting needs the following

- (a) Simple classifiers to be combined.
- (b) Training error rate of these classifiers.

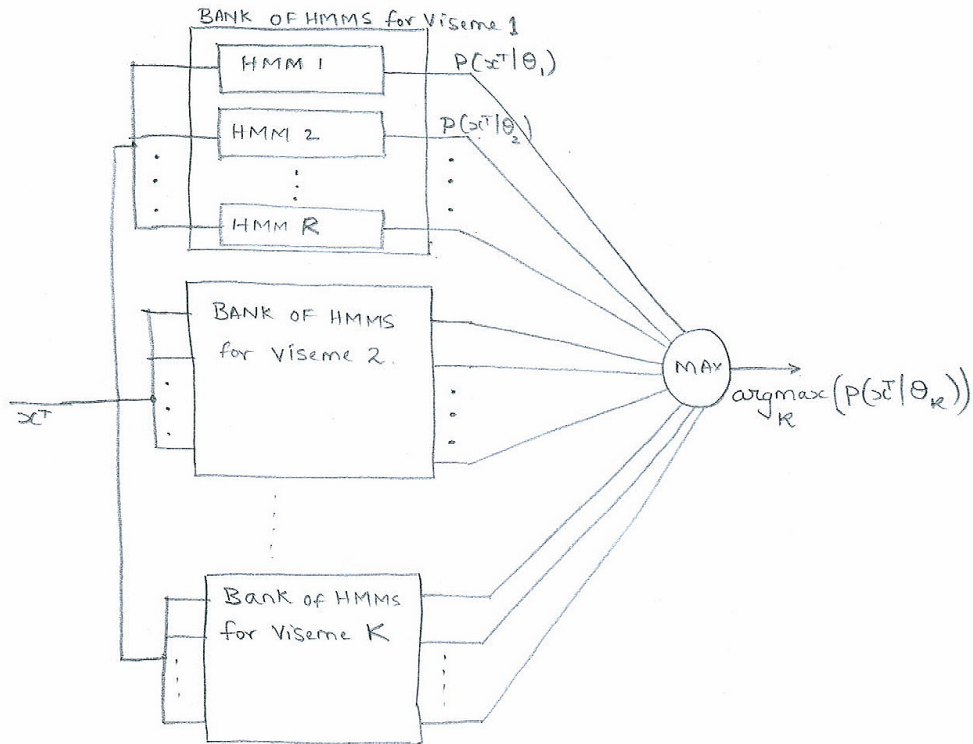
For class k , start with a weighted training set.

Train the HMM θ_k on this weighted training set, using Baum-Welch recursion.

The HMM gives a probability measure and not a hard decision. The hard decision is made by comparing with all the HMMs of other classes.

$$h_k(x) = \begin{cases} 1 & \text{if } P(x|\theta_k) > P(x|\theta_j) \\ -1 & \text{otherwise.} \end{cases} \quad j \neq k$$

Multiclass Boosting



Boosting HMMs

While boosting, each class will have multiple HMMs.

The comparison is over all the HMMs belonging to other classes.

$$h_k(x) = \begin{cases} 1 & \text{if } P(x|\theta_k) > P(x|\theta_j) \\ -1 & \text{otherwise.} \end{cases} \quad j \neq k$$

The error rate for this classifier is given by

$$\epsilon_k = \sum_{\substack{i=1 \\ h_k(x_i)=-1}}^R D(x_i)$$

The error rate depends not only on the classifier θ_k but also on all other classifiers.

In boosting round t , the new HMM might change the error rate of the previously added HMMs of the other classes.

Error computation

Efficient error rate computation can be made by simple book-keeping.
For each training sample of class k compute and store

$$P_{\max}(x_i|\bar{\theta}) = \max_{\theta_j} P(x_i|\theta_j) \quad j \neq k$$

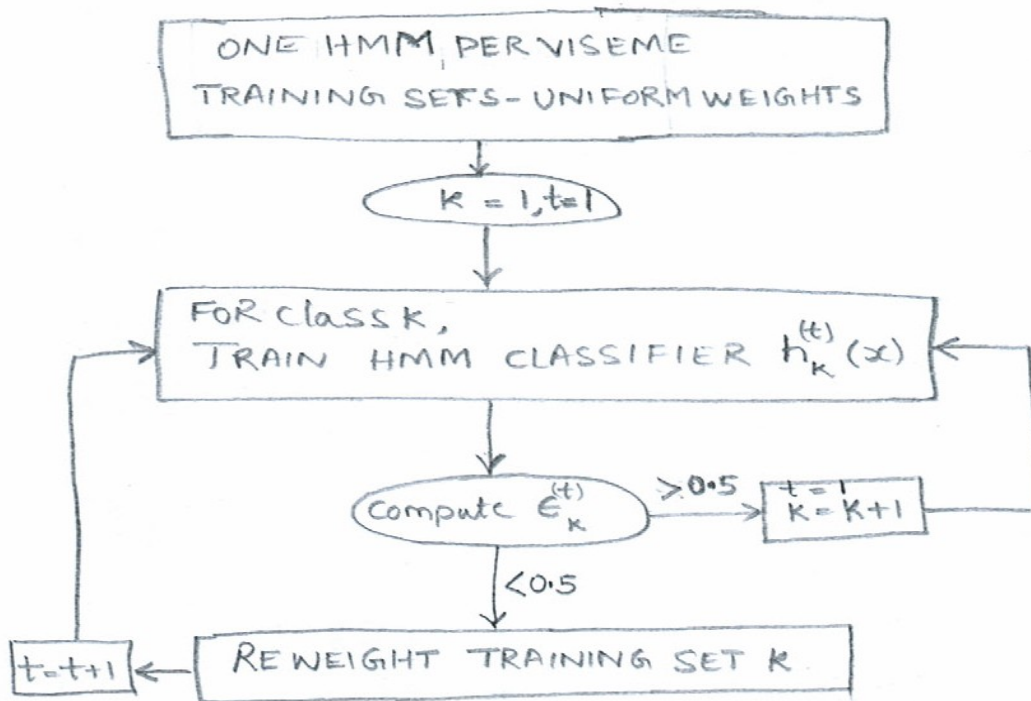
When a new HMM is added to class k , the $P_{\max}(x_i|\bar{\theta})$ is updated for all training samples of all the other classes.

The HMM classifier can now be written as

$$h_k(x) = \begin{cases} 1 & \text{if } P(x|\theta_k) > P_{\max}(x|\bar{\theta}) \\ -1 & \text{otherwise.} \end{cases}$$

Boosting can continue if the error rate of all the individual HMMs is less than 0.5.

AdaBoost for Multiclass Classifier Banks



AdaBoost for Multiclass Classifier Banks

(1) Start with uniformly weighted training sets for each class and one classifier per class. Start boosting class 1.

(2) For the class being boosted, train a new classifier θ_t^k with the weighted training set.

(3) If error rate ≤ 0.5 for all classifiers, go to step (4). Otherwise start boosting class $k+1$. Goto step (2).

(4) Calculate

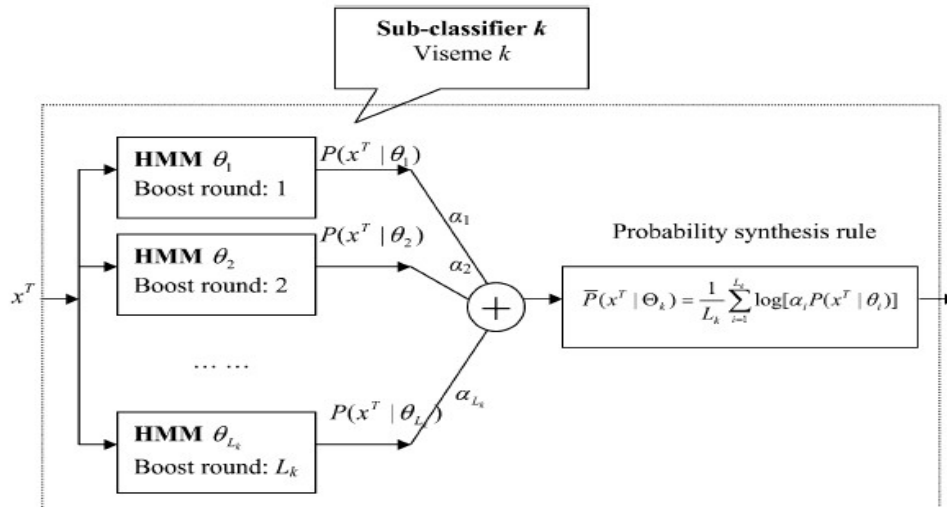
$$w_t^k = \frac{1}{2} \ln \frac{(1 - \epsilon_t^k)}{\epsilon_t^k}$$

If error rate of some other classifier has changed, recompute its weight.

(5) Update the training set weights. Goto step (2).

Combiner

The final decision is given by



Comment : The boosted classifier in the AdaBoost algorithm is of the form $H_k(x) = \text{sgn} \sum_t w_t \cdot h_t(x)$. This is not the same as combining probabilities.

Comments

The error used in the boosting algorithm is only part of the classification error.

Only false negatives (Type 2 errors) are considered.

False positives or Type 1 errors are not considered.

This is because the HMM is trained with positive examples only.

Results

In the case of isolated visemes, the classification error for the single and boosted HMMs are shown below.

TABLE II
CLASSIFICATION ERRORS (FRR) OF THE SINGLE-HMM CLASSIFIER AND THE
ADABOOST-HMM CLASSIFIER IN RECOGNITION OF ISOLATED VISEMES

Viseme Categories	Classification Error (FRR)	
	Single-HMM Classifier	AdaBoost-HMM Classifier
1 p, b, m	13%	8%
2 f, v	4%	5%
3 T, D	11%	4%
4 t, d	35%	17%
5 k, g	24%	9%
6 tS, dZ, S	10%	10%
7 s, z	4%	4%
8 n, l	19%	20%
9 r	18%	7%
10 A:	1%	4%
11 e	8%	11%
12 I	1%	4%
13 Q	7%	10%
14 U	7%	9%
Average	11.6%	8.7%

Results

In the case of context-dependent visemes, the training error and classification error for the single and boosted HMMs are shown below.

TABLE III
TRAINING ERRORS AND CLASSIFICATION ERRORS (FRR) OF THE SINGLE-HMM
CLASSIFIERS AND THE ADABOOST-HMM CLASSIFIERS

Viseme Categories	Single-HMM Classifier		AdaBoost-HMM Classifier	
	Training Error	Classification Error	Training Error	Classification Error
1 p, b, m	14%	20%	6%	15%
2 f, v	15%	27%	11%	25%
3 T, D	24%	34%	10%	18%
4 t, d	39%	50%	17%	19%
5 k, g	17%	17%	16%	16%
6 tS, dZ, S	17%	21%	5%	9%
7 s, z	39%	45%	13%	17%
8 n, l	40%	79%	22%	33%
9 r	21%	54%	22%	37%
10 A:	14%	18%	5%	5%
11 e	27%	33%	13%	7%
12 I	9%	10%	0%	2%
13 Q	10%	35%	4%	11%
14 U	4%	9%	1%	7%
Average	21%	32%	10%	16%

Discussion

1. General algorithm for boosting bank of classifiers, not just HMMs
2. Analysis of the results
3. Loopholes in algorithm.

References

- [1] Say Wei Foo, Yong Lian, Liang Dong, *Recognition of Visual Speech Elements Using Adaptively Boosted Hidden Markov Models*, IEEE Transactions on Circuits And Systems for Video Technology, Vol. 14, No. 5, May 2004.
- [2] L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition* , Proc. IEEE, vol. 77, pp. 257286, Feb. 1989.
- [3] Tsuhan Chen, *Audio Visual Speech Processing*, IEEE Signal Processing Magazine, January 2001.