

# Speaker Recognition using keyword Hidden Markov Models and Support vector machines

Author: Howard Lei (SID: 17164723)

Affiliation: International Computer Sciences Institute (ICSI)

## I. Speaker recognition overview

Speaker recognition has traditionally relied on low-level acoustic features from speech signals. This has typically been done using a text-independent, bag-of-frames approach, where each signal frame is treated separately and independent of other frames as well as speech text. Gaussian mixture models are typically built from acoustic features, and used for classification of speakers.

New approaches to speaker and background model training have given rise to many recent developments in speaker recognition. Recently, various text-dependent approaches have surfaced, including a keyword Hidden Markov Models (HMM) approach [1]. This approach also deviates from the traditional bag-of-frames approach by taking into account relationships in time among acoustic features for different signal frames. In many of these text-dependent approaches, acoustic features are obtained not for the entire speech signal, but for only parts of the signal corresponding to certain words. The words are usually chosen rather arbitrarily – usually based on perceived frequency and variability in pronunciation. The belief is that the different ways that people pronounce these words will provide enough speaker discriminative information.

This paper discusses two text-dependent speaker recognition approaches that were implemented. The first is the traditional keyword HMM approach (which will act as a baseline for comparison), while the second is a new approach using support vector machine (SVM) models trained using data from HMMs.

## II. MFCC feature extraction

The acoustic features used throughout the experiments are Mel-Frequency Cepstral coefficients (MFCC) [2]. These features are widely used in both speaker and speech recognition, and a general procedure for their extraction, along with some explanations, will be given.

The speech signal is first segmented into 25 ms frames (with Hamming window applied) with 15 ms overlap (and hence with a 10 ms sampling period), and MFCC feature extraction is performed separately for each frame. It is widely known that the frequency content of speech signals can provide valuable information with regards to pitch, smoothness, and other perceived qualities of a signal. In particular, because the human ear hears frequencies spaced on a mel-scale, extracting frequencies on a mel-spaced frequency scale from the spectrum of a signal allows one to capture important characteristics of speech as perceived by humans [2].

Hence, to extract MFCC features, a mel-scaled filterbank (a bank of band-pass filters) is first applied to the signal, and the energies within each band are used for MFCC extraction [2]. In particular, the log energies of the signal within those bands are obtained, and the discrete cosine transform (DCT) of those log energies is performed to un-correlate the log energies, projecting them onto maximum variance dimensions (this is similar to the PCA technique, except that instead of projecting data onto the eigenvectors, data is projected along the dimensions spanned by the rows of a DCT matrix) [2]. These projected and de-correlated log energies are the MFCCs. MFCCs have been shown to be a very effective acoustic feature for speech and speaker recognition [3]. Fig. 1 illustrates the feature extraction procedure.

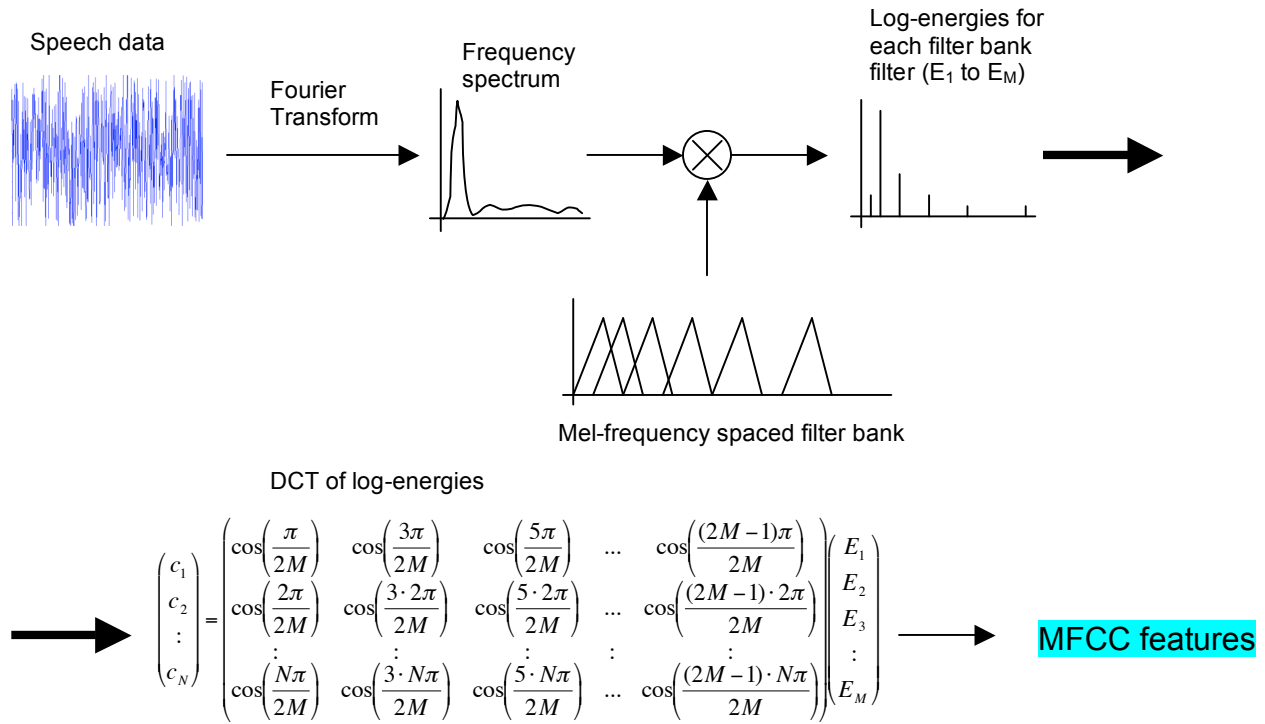


Figure 1. MFCC feature extraction

In these experiments, the HTK tool is used for MFCC feature extraction [4]. 20 MFCCs (along the dimensions spanned by the first 20 rows of the DCT matrix) are used as features, and the differences between each of the 20 MFCCs in the current signal frame and an adjacent frame are also used as features. Hence, for each frame, a 40-dimensional feature vector is obtained. Because the frame sampling rate is 10 ms, the 40-dimensional feature vectors of adjacent frames are from a time difference of 10 ms.

### III. Speaker recognition using traditional keyword HMM approach

#### i. Terminology and basics

It will be helpful for the reader to have definitions of some commonly used terminology.

Term	Definition
Conversation side	speech data containing 2.5 to 5 minutes of speech from a single speaker
Target Speaker	a speaker that a conversation side is tested for
Test utterance	a conversation side that is tested for various target speakers
Background speaker	a speaker that is not a target speaker
Keyword	a single word, or a 2 word sequence
Keyword HMM	a Hidden Markov Model trained using speech data from only one keyword

The traditional HMM speaker recognition approach using 19 keywords was first implemented by Boakye [1]. In these experiments, a total of 39 keywords will be used, and a separate result will be obtained for each keyword (this was not performed by Boakye, who looked only at the performance for all keywords combined together). These results obtained will be used as a means of comparison for the new SVM approach.

The parameters of HMMs are trained using MFCC features from portions of speech signals corresponding to 39 different keywords: *about, actually, all, anyway, because, but, have, i\_know, i\_mean, i\_see, i\_think, just, know, like, mean, no, not, now, okay, one, people, really, right, see, so, that, there, think, this, uh, uhhuh, um, was, well, what, yeah, yep, you\_know, you\_see*. Transcriptions of speech

signals, obtained from Stanford Research Institute (SRI), are used to identify portions of speech signals belonging to those keywords. Note that some words (i.e. you\_know, you\_see, i\_see, etc) are actually a sequence of two words. The words are chosen based on both perceived variability of pronunciation, and frequency of appearance in the transcriptions of a sizable portion of the speech data. The key to this speaker recognition approach is to capture the time-dependent acoustic feature information of different speakers for each of the keywords.

*i. HMM Training*

One HMM is first trained using the HTK software [4] for each keyword (referred to as a keyword HMM) using speech data from all background speakers. This speech data is comprised of 1553 conversation sides, obtained from NIST's 2003 evaluation and Fisher corpus. MFCCs from every instance of a given keyword from every background conversation side is used to train the background keyword HMM.

Each HMM consists of 8 40-dimensional (due to the 40-dimensional feature vector) Gaussian mixture components for each state. They are left-to-right with self-loops at each state, but can not skip states. The number of states for each keyword HMM is derived from the number of frames and phonemes for the keyword:

$$\text{NumStates} = \min(3 * P, 1/4 * D)$$

where P is the number of phonemes comprising the keyword, and d is the median number of signal frames of the keyword

Keyword HMM parameters are trained with the background speaker data using the Expectation-Maximization (EM) algorithm. Fig. 2 illustrates the EM algorithm as used by HTK.

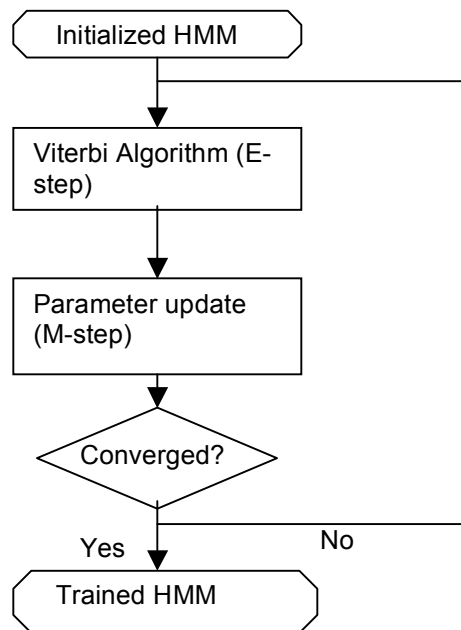


Figure 2. EM algorithm used by HTK

To begin keyword HMM parameter training, the parameters are first initialized. Then, the Viterbi algorithm is used to give an estimate of the sequence of states given the sequence of feature vectors belonging to an instance of a keyword (E-step). The keyword HMM transition probability matrix and observation distribution parameters are then updated, attempting to maximize the probability that the particular sequence of states was obtained from the transition probability matrix, and the probability that the sequence of feature vectors was observed from the given sequence of states (M-step). These two steps are repeated until convergence. The keyword HMM trained using background speaker data will be referred to as the background keyword model.

Keyword HMMs for each target speaker (otherwise known as the target speaker keyword models) will be obtained from the background keyword model. Specifically, the keyword HMM parameters for a target speaker are obtained via MAP adaptation of the Gaussian means of each HMM state from the background keyword model. This ensures a certain uniformity between the background HMM and each target speaker HMM of a keyword. In particular, the number of HMM states is guaranteed to be equal for all HMMs (background and target speaker models) of a certain keyword.

Each target speaker keyword model is trained using 8 conversation sides of data from that speaker. MFCC data from all instances of the keyword in those conversation sides are used for keyword HMM training. The target speaker conversations sides are from NIST's 2005 evaluation corpus (SRE05). A total of 3696 SRE05 conversation sides are used for target speaker keyword model training, for a total of 464 target speaker models. Note that a couple target speakers have slightly fewer than 8 conversation sides, due to bad speech data released by NIST, and/or bad transcriptions of speech data. This should have negligible effect on the overall results, however.

MAP adaptation using HTK is accomplished by the following equation for state  $j$  and Gaussian mixture  $m$  [4]:

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm} + \tau} \bar{\mu}_{jm} + \frac{\tau}{N_{jm} + \tau} \mu_{jm}$$

where  $\tau$  is the weight of the a priori knowledge (the Gaussian mean from the background keyword model) to the adaptation data (the Gaussian mean from the adaptation data),  $N_{jm}$  is the occupation likelihood of the adaptation data,  $\bar{\mu}_{jm}$  is the Gaussian mean of the adaptation data,  $\mu_{jm}$  is the Gaussian mean of the background keyword model, and  $\hat{\mu}_{jm}$  is the updated Gaussian mean.

Note that only the means of all HMM states, excluding the first and last states, are adapted for each target speaker keyword model; the transition probabilities, observation probability distributions, and Gaussian variances are kept the same. If a keyword does not appear in the conversation sides of a target speaker, the target speaker keyword model will simply be copied from the background keyword model. This ensures the existence of a target speaker model for each keyword, although a couple may be the same as the background keyword model.

Fig. 3 illustrates the MAP adaptation of Gaussian means of the background keyword model given the MFCCs for a keyword from conversations sides of a target speaker.

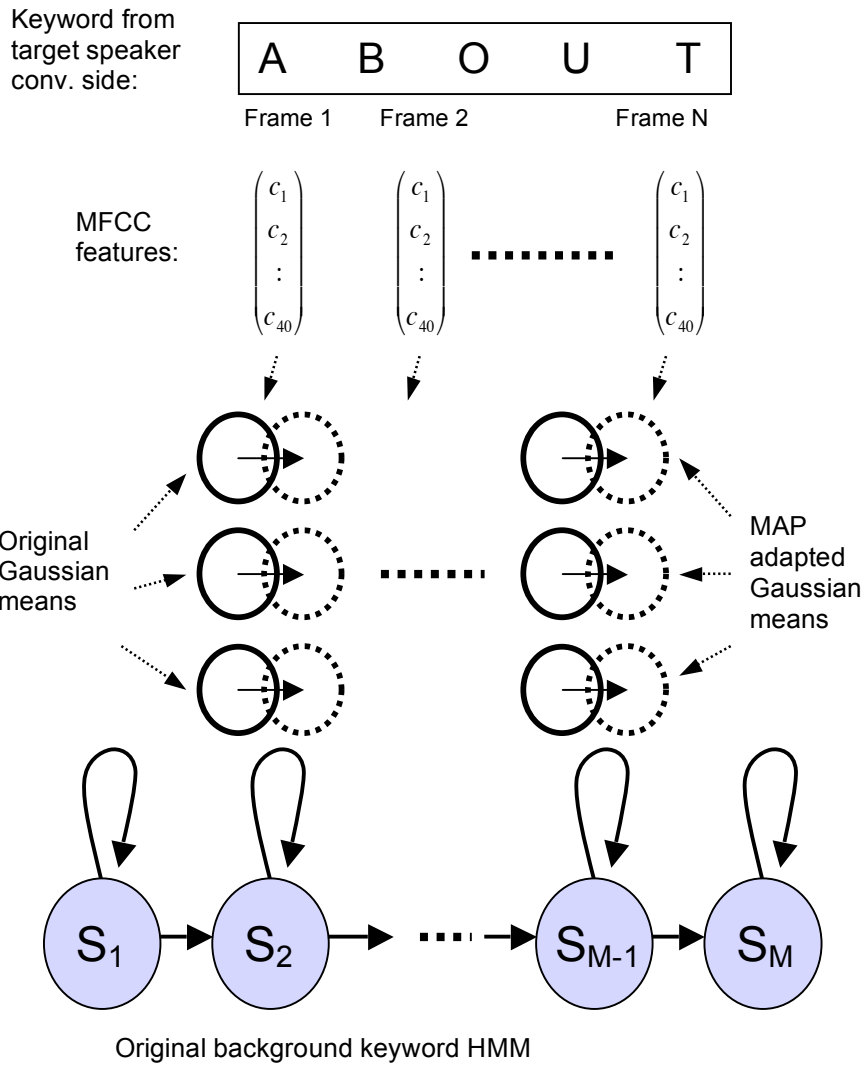


Figure 3. MAP adaptation of background keyword model to create target speaker keyword model

ii. *HMM Testing*

For testing, the sequence of feature vectors  $(C_1, \dots, C_N)$  belonging to a keyword instance in a test utterance conversation side is scored against a subset of target speaker keyword models. The score is obtained via the following log-likelihood ratio:

$$\log \left( \frac{p(C_1, \dots, C_N | M_{TS})}{p(C_1, \dots, C_N | M_{BKG})} \right)$$

where  $M_{TS}$  is the target speaker keyword model,  $M_{BKG}$  is the background keyword model, and

$$\log(p(C_1, \dots, C_N | M)) = \log \left( \sum_{path} p(C_1, \dots, C_N | path, M) p(path | M) \right)$$

Because there may be multiple instances of a keyword in a test utterance conversation side, the log-likelihood ratio is obtained for each keyword instance, and summed over all keyword instances. This log-likelihood ratio sum is then divided by the total number of frames in all keyword instances of a test utterance conversation side to arrive at the final keyword score for a given test utterance tested against a target speaker model. In order to combine the keywords such that a single score is obtained for each test utterance and target speaker model pairing, the log-likelihood ratios of all instances of all keywords are added, and divided by the total number of frames in all instances of all keywords. Fig. 4 is a summary illustration of the traditional keyword HMM approach.

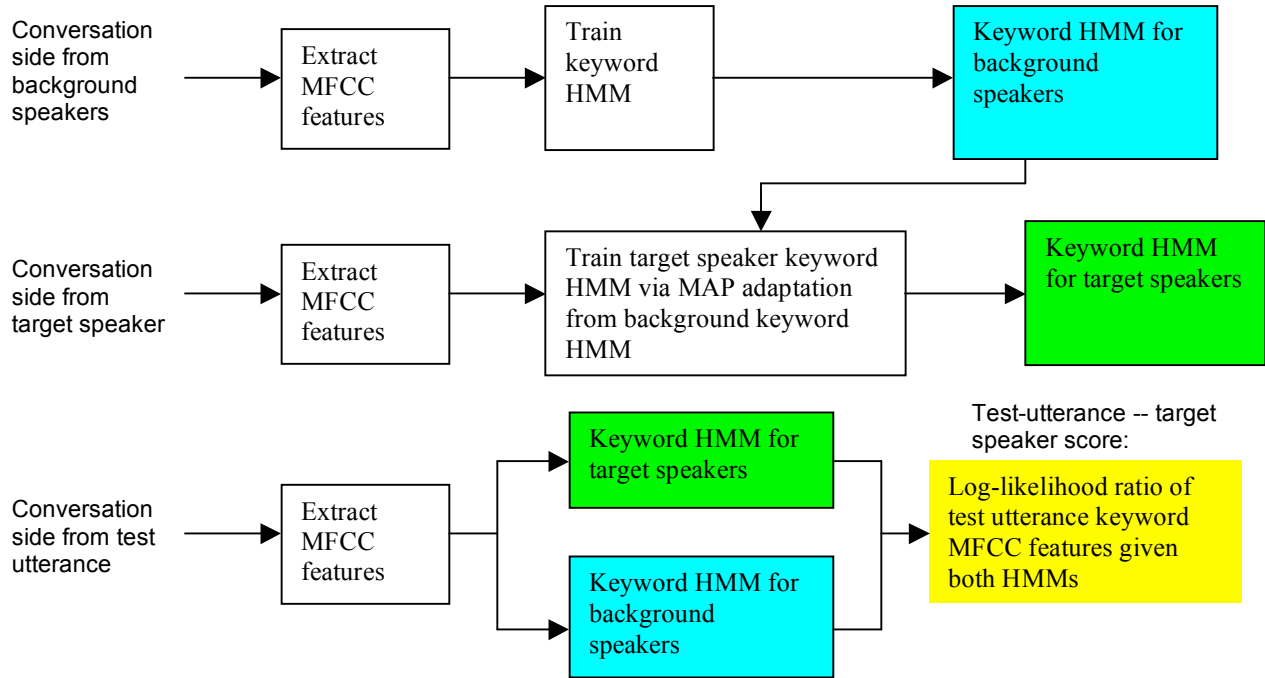


Figure 4. Speaker recognition paradigm using traditional HMM approach

Test utterance conversation sides are also from the SRE05 corpus, separate from the target speaker conversation sides. A total of 2283 test utterance conversation sides are used.

### iii. Results

Each test utterance is tested against a subset of the speaker models, resulting in a total of 20825 scores (the total number of test utterance – target speaker pairings). For each pairing, if the target speaker spoke in the test utterance, the score for that particular pairing is a true speaker score; otherwise it's an impostor score. Given all the scores, a scoring threshold can be established to separate the true and impostor scores. A false accept (FA) occurs when an impostor score is classified by the threshold as a true speaker score; a miss (MI) occurs when a true speaker score is classified as an impostor score. The threshold can be varied to give different error rates. The equal error rate (EER) occurs at a threshold at which FA equals MI. This EER is used to characterize the effectiveness of the over speaker recognition approach. The lower the EER, the better the approach. This is the standard NIST scoring procedure.

Table 1 shows the results of the traditional HMM approach for each of the 39 keywords, sorted in terms of EER:

Keyword	EER (%)	# of HMM states	# of instances in background conversation sides
you_know	12.03	8	17349
yeah	12.35	6	26530
right	15.05	8	8021
um	15.33	6	11962
i_think	15.43	9	6288
that	15.77	5	26277
like	15.84	5	18058
because	16.02	8	5164
i_mean	16.31	7	5470
but	16.56	5	12766
have	17.43	5	9610
just	18.06	6	8660
people	18.06	8	4906
so	18.37	6	14291
really	18.46	7	6674
not	18.56	5	6817
think	20.36	6	3179
about	20.37	7	5769
uhhuh	21.56	9	8371
uh	21.59	3	18065
now	23.07	6	2851
okay	24.19	9	4322
this	24.72	5	5408
was	24.86	5	9888
i_know	25.81	9	2142
one	25.83	5	4559
there	26.58	5	4716
what	26.71	5	8088
no	27.01	6	4245
know	27.96	5	4767
all	28.17	5	4681
actually	28.17	9	2240
see	30.33	6	2006
well	31.10	6	7590
anyway	41.76	10	437
yep	50.00*	8	211
mean	50.00*	5	254
you_see	50.00*	9	247
i_see	50.00*	10	505

\* there are too few instances of these words

Table 1. Keyword results for traditional HMM approach

Note that in the event that a keyword with a 2 word sequence contains another keyword, only the keyword with the 2 word sequence is counted. According to these results, the keyword you\_know provides the most speaker discriminative power. There appears to be a general correlation between the EER and the number of keyword instances. This shows that the more training examples for a keyword, the better the HMM performance. There does not appear to be a correlation between the EER and the number of HMM states for the keyword.

If the keywords are combined via the aforementioned procedure, the resulting EER is 6.08%, which is slightly higher, but comparable to some of the top speaker recognition approaches at ICSI [6]. Note that the EER for the combined experiment is significantly lower than the EER for any of the keywords.

#### IV. Speaker recognition using anchor models, HMM super-vectors, and support vector machines.

This is the new speaker recognition approach, based on a very similar approach involving GMMs by Campbell et. al [7]. In this new approach, the means of the Gaussian mixture components of keyword HMMs are used as feature vectors in a support vector machine (SVM) classifier. Each target speaker's MAP adapted keyword HMM will result in a different set of Gaussian means, which themselves may hold enough speaker discriminative power. These Gaussian means will be from keyword HMMs trained the same way as before, and will be used as high-dimensional feature vectors for SVM training and classification. One SVM model will be trained for each target speaker. Hence, data from each target speaker will act as positive training examples, while data from a set of non-target speakers (anchor speakers) will act as negative training examples.

##### *i. Terminology*

Before explaining further, it would be helpful to the reader to define some important terminology.

Term	Definition
Anchor speakers	Speakers that are neither target nor background speakers (note that the term anchor models has been used in the past by Sturim and Reynolds et. al in a different formulation [8])
Supervector [7]	A vector of the means of all the Gaussian mixture components of all states excluding the first and last states in a keyword HMM. (The first and last states are excluded because they are unchanged through MAP adaptation.)
Test-utterance model	A keyword HMM adapted for each test utterance conversation side. Note that only a single conversation side is used for this adaptation. Also note that not all keywords are likely to appear in only a single conversation side. If a keyword does not appear, the adapted keyword HMM will be replaced by the background keyword model

##### *ii. Anchor speaker keyword HMM training*

Anchor speaker keyword HMMs are trained in exactly the same manner as target speaker HMMs, via MAP adaptation from the background keyword HMMs. A total of 1330 anchor speaker keyword models are trained (using 8 conversation sides each), 1105 of which from NIST's 2003 evaluation corpus, and 225 from NIST's 2004 evaluation corpus. Note that the speaker models in these corpora are different from those in NIST's 2005 evaluation corpus, which are used to train target speaker keyword models. This ensures that all anchor speaker models will be from non-target speakers.

The Gaussian mixtures means from every state excluding the first and last states of an anchor model keyword HMM will be collected in a supervector. As stated previously, each HMM state contains 8 Gaussian mixture components, each of 40 dimensions (due to the 40-dimensional acoustic feature vectors). Hence, for a keyword HMM with 7 states (5 excluding the first and last states), a 1600 dimensional supervector is obtained. Note that the number of keyword HMM states depend only on the keyword. The 1300 anchor keyword model supervectors form one large negative data class for SVM training against supervector(s) from target speaker keyword models. Hence, anchor speaker supervectors act as negative training examples, and target speaker supervectors act as positive training examples.

##### *ii. Target-speaker keyword HMM training, revisited*

The target speaker keyword model training approach is the same as before (using MAP adaptation). However, the previous training approach trains only one keyword HMM for each target speaker, such that only one positive training example supervector is obtained for each target speaker. This is slightly disconcerting, but due to the limited amount of available NIST data, it is not feasible to train different keyword HMMs for the same target speaker on multiple sets of 8-conversation side data (thus resulting in more than one positive training example supervector).

The approach used in some of the following experiments is to take 8 different subsets of the 8 conversation sides of a target speaker, and train a target speaker keyword HMM using each subset. Hence, a total of 8 keyword HMMs will be obtained, resulting in 8 positive training examples for each target speaker SVM model.

One SVM model is trained for each target speaker against the 1300 anchor speakers. Fig. 5 illustrates SVM training for a given target speaker M.

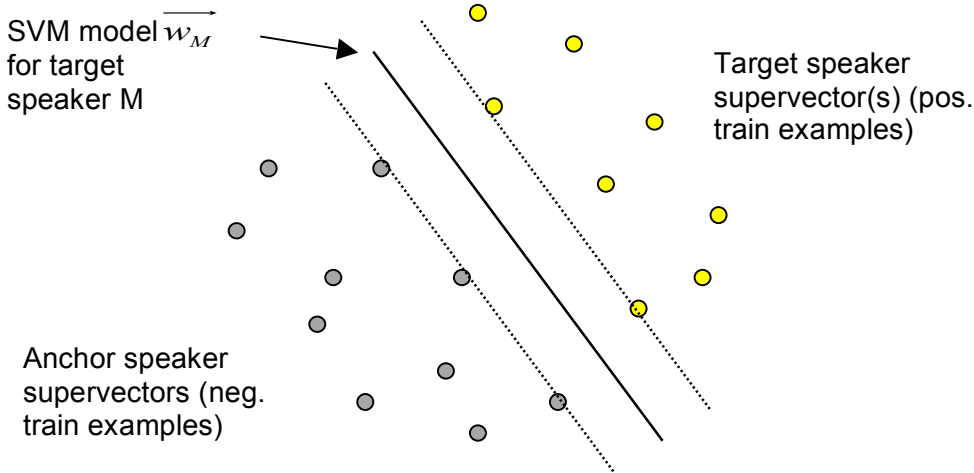


Figure 5. Target speaker SVM model training

For these experiments, the SVM model will be trained using a linear kernel. The SVM hyperplane  $\vec{w}_M$  is obtained as follows:

$$\min_{\vec{w}_M, b_M, \xi_i, \eta_j} \left( \frac{\|\vec{w}_M\|^2}{2} + C \sum_i \xi_i + D \sum_j \eta_j \right)$$

$$\begin{aligned} \text{s.t. } & \vec{w}_M \cdot \vec{t}_i + b_M \geq 1 - \xi_i & \forall i \\ & -(\vec{w}_M \cdot \vec{a}_j + b_M) \geq 1 - \eta_j & \forall j \\ & \xi_i \geq 0 & \forall i \\ & \eta_j \geq 0 & \forall j \end{aligned}$$

where  $\vec{t}_i$  represents a target speaker supervector,  $\vec{a}_i$  represents an anchor model supervector,  $\xi_i$  is the training error for each positive example, and  $\eta_i$  is the training error for each negative example. Note that different weights can be assigned to positive and negative training errors using the C and D parameters. All SVM training and testing is done using the SVM<sup>light</sup> software package [5].

iii. *Test-utterance keyword HMM training and SVM testing*

For testing, a keyword HMM will be trained for each test utterance using the same MAP adaptation approach as before. The test utterance supervector  $\vec{x}$  obtained from its keyword HMM will be scored using the SVM models trained for the target speakers:

$$Score = \vec{w}_M \cdot \vec{x} + b_M$$

Hence, a more positive score will likely be assigned to a test utterance – target speaker pairing if the target speaker spoke in the test utterance (since their supervectors will more likely be on the same side of the hyperplane), and vice versa. One score will be assigned for each test utterance tested against each target speaker (as was done previously). The same thresholding technique as previously described is used again, and the equal error rate (EER) is once again used to characterize the overall performance of the approach.

Note that the above is done for each keyword separately, such that an EER is obtained for each keyword. To combine the keywords, the supervectors obtained from each target speaker HMM, anchor speaker HMM, and test utterance HMM for each keyword are concatenated into much higher dimensional supervectors. For instance, the supervectors for target speaker A for keyword 1 are concatenated with the supervectors for target speaker A for keywords 2, 3, 4, ..., 39. This is repeated for the anchor speaker and test utterance supervectors. Hence, if there are 39 keywords, and each keyword supervector has length 1600 (hypothetically), the dimension of each concatenated supervector will be 62,400. The same SVM training and classification is then performed on these higher dimensional supervectors, and an EER is obtained for this keyword combined approach.

iv. *Results*

Using the standard NIST scoring procedure, each test utterance is tested against a subset of target speakers, resulting in a total of 20825 scores. The SVM training for each keyword is such that a positive example training error weighs 50 times as much as a negative example training error (C/D = 50). This is due to the vastly greater amounts of negative training examples.

Table 2 shows the results of the new approach for each of the 39 keywords, sorted in terms of EER. For these results, all 8 target speaker conversation sides are used to train the keyword HMM for each target speaker, from which the supervectors are obtained. Thus, there is only one keyword HMM per target speaker, and only one positive training example is used to train each target speaker SVM model.

Keyword	EER (%)	Dimension of super-vector	# of instances in background conversation sides
you_know	17.23	1920	17349
yeah	18.00	1280	26530
that	19.59	960	26277
um	22.39	1280	11962
like	22.92	960	18058
but	23.26	960	12766
uh	24.23	320	18065
have	25.92	960	9610
right	26.35	1920	8021
so	26.59	1280	14291
because	27.90	1920	5164
not	28.72	960	6817
i_think	29.05	2240	6288
just	29.58	1280	8660
uhhuh	30.50	2240	8371
i_mean	31.32	1600	5470
people	32.29	1920	4906
what	32.82	960	8088
about	33.11	1600	5769
really	34.56	1600	6674
was	35.32	960	9888
well	35.91	1280	7590
this	36.10	960	5408
one	36.10	960	4559
no	36.15	1280	4245
all	36.34	960	4681
think	37.16	1280	3179
know	37.36	960	4767
okay	37.40	2240	4322
there	37.98	960	4716
now	40.73	1280	2851
see	41.36	1280	2006
i_know	41.36	2240	2142
actually	45.51	2240	2240
i_see	46.09	2560	505
mean	47.15	960	254
anyway	47.25	2560	437
yep	50.00	1920	211
you_see	50.00	2200*	247

\*This should be a 2240 dimensional super-vector due to its 10 HMM states. The 2200 dimensions is due to a HTK software bug resulting from the severe lack of training data. This should have negligible effect on the following results, however.

Table 2. Keyword results using new SVM approach

According to these results, the individual keyword EERs are not as good as they were in the traditional HMM approach. However, keywords that performed well in the previous approach also performed relatively well in this new approach.

Table 3 shows the keyword combined results. Different numbers of conversation sides are used to train target speaker models, and different weights of positive training example errors to negative training example errors are experimented with.

Number of target speaker model conversation sides	Weight of pos. example training error to neg. example training error (C to D ratio)	EER (%)
8 (1 pos. training example)	500	5.93
8 (1 pos. training example)	50	5.50
8 (1 pos. training example)	1	5.98
7 (8 pos. training examples)	500	5.89
7 (8 pos. training examples)	50	5.79
7 (8 pos. training examples)	1	5.41
5 (8 pos. training examples)	500	5.94
5 (8 pos. training examples)	50	5.85
5 (8 pos. training examples)	1	4.88
3 (8 pos. training examples)	500	6.72
3 (8 pos. training examples)	50	6.18
3 (8 pos. training examples)	1	4.54

Table 3. Keyword combined results using new SVM approach

These results show that the best results for combination of the keywords using the new approach (EER of 4.54%) is better than the combination result (EER of 6.08%) for the traditional HMM approach. Taking subsets of the 8 target speaker conversation sides to create 8 target speaker model supervector data points for SVM training can improve results. In the case where only 3 conversation sides are used to train a HMM and to create a supervector, the results are worse (6.72% EER) with C/D being 500 and 50, but become surprisingly good with C/D being 1. This could be that the data classes spanned by the target speaker supervectors for HMMs trained using only 3 conversation sides are more spread out, perhaps overlapping with the data class spanned by the anchor model supervectors. If so, then having positive example training errors is acceptable. Note that these results are comparable to some of the top speaker recognition approaches at ICSI [6].

## VI. Conclusion and future work

In this paper, two main approaches involving HMMs are illustrated for purposes of speaker recognition. The first approach is the traditional approach that trains HMMs using acoustic feature vectors from different frames of a speech signal, and directly uses the HMMs to perform classification. Classification of a test utterance is performed using HMMs trained on target speaker models, and a background model. The log-likelihood ratio of a sequence of acoustic feature vectors is used to score a test utterance against a target speaker. In the second approach, the MAP adapted Gaussian means of the HMMs for target speakers, anchor speakers, and test utterances are gathered into a supervector. A SVM model for each target speaker is trained using these supervectors. The SVM model then classifies super-vectors from test utterances to score the test utterance speaker against the target speaker.

There are many directions that this research can take. One possible path could be to determine a better set of keywords to use and combine. Because these experiments are computationally expensive, it would be very advantageous to have a very small subset of keywords that provide good speaker discriminative power via the HMM approaches. Another path could be to find better HMM models for the keywords. Different types of kernels could be investigated for the new SVM approach. In short, there are countless approaches one could take to improve these speaker recognition methodologies.

## References

- [1] K. Boakye, "Speaker Recognition in the Text-Independent Domain Using Keyword Hidden Markov Models." unpublished master's thesis, ICSI, 2005
- [2] S.B. Davis, P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 4, ASSP-28, No. 4, pp. 357-366, 1980.
- [3] D. Reynolds, "Experimental Evaluations of Features for Robust Speaker Identification," IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 4, pp. 639-643, 1994.
- [4] HMM Toolkit (HTK): <http://htk.eng.cam.ac.uk>
- [5] SVM<sup>light</sup> software: <http://svmlight.joachims.org/>
- [6] N. Mirghafori et. al, "ICSI's 2005 Speaker Recognition System."
- [7] W.M. Campbell et. al, "SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP Variability Compensation," in proceedings of ICASSP, Vol. 1, pp. I-97 – I-100, 2006.
- [8] Sturim et. al, "Speaker Indexing in Large Audio Databases Using Anchor Models," in proceedings of ICASSP, Vol. 1, pp. 429-432, 2001.