

Speaker Dependent Activation Keyword Detector Based on GMM-UBM

Evelyn Kurniawati, Sapna George

STMicroelectronics Asia Pacific Pte. Ltd.

evelyn.kurniawati@st.com, sapna.george@st.com

Abstract

In this paper, we present a new method for isolated keyword detection that is meant to activate a personal device from standby state. Instead of using the common method for speech recognition such as Hidden Markov Model (HMM) or Dynamic Time Warping (DTW), we modify a GMM-UBM (Gaussian Mixture Model – Universal Background Model) scheme that is better known in speaker recognition field. Since only one adapted Gaussian mixture is used to represent the keyword, a second layer of check is employed to ensure the right sequence of occurrence within the keyword. This is done by comparing it with the Longest Common Subsequence (LCS) of the highest performing GMM component obtained during the registration phase. Results for a subset of the SpeechDat-Car database are presented to validate the benefit of this modeling against moderate noise level.

Index Terms: keyword spotting, GMM-UBM

1. Introduction

The field of speech and speaker recognition has been progressing steadily these past few years [1][2]. Services such as voice command, voice biometric, and even personal assistance that understands natural speech is gaining acceptance and have become a product differentiator. Which technology to choose for a particular system mostly depends on the functionality required as well as the memory footprint and complexity limitation.

This paper deals with a specific application which allows an authorize user of a device to activate or wake up the device from standby mode. Since it is going to run during a standby mode, it is important for it to be low in complexity. Furthermore only the user of the device can perform the activation, so the solution needs to have a speaker verification element.

There are two main methods in speech recognition, statistical based using HMM and template matching based with DTW. HMM requires many transcribed speech data to build the statistical model. DTW on the other hand, only require at least one speech sample of the word to be recognized. The complexity and database size of DTW system however, is proportional to the number of template words that are stored. Furthermore, DTW system is less flexible in modeling the variability of speech.

In text independent speaker recognition, GMM-UBM is one of the successful approaches. Its combination with Support Vector Machine [3] yields excellent results in NIST (National Institute of Standards and Technology) Speaker Recognition Evaluation [4]. The idea is to build a GMM that represent a single speaker by performing adaptation from Universal Background Model that was trained from various speakers. During the testing phase, the UBM is used to normalize the score. The test speech is considered to belong to the target speaker if there is a significant ratio between the likelihood generated from the speaker model and the UBM.

This work takes cue from the above methodology. However, in this case the UBM is made to represent all the speech from the user. This speech can be obtained in inconspicuous manner, for example while the user is using the device. During the training phase when a particular word is given as the keyword, this keyword is used to adapt the UBM resulting in a model for that particular keyword. Similar normalization scheme is adopted, whereby a test speech is considered to be that keyword if there is a significant ratio between the likelihood from that keyword model and the UBM.

One problem with the above scheme is that while there is no temporal axis needed for a speaker recognition task, it is necessary for the speech recognition task. For example, the solution described above would not be able to distinguish between the word “My TV” and “TV My”, because the word model is represented by a flat GMM. This is where we introduce an extra check. If a test speech is considered to be likely to come from the keyword model, a synchronization point comparison is performed with the keyword template. Unlike the traditional DTW which uses the MFCC as templates, this work only uses the index of the best performing component of the keyword model. This significantly reduces the space requires to store the template. Furthermore, only one template is stored in the system by performing the Longest Common Sequence (LCS) analysis on the registration word. In other words, the single template represents the longest sequence of highest performing GMM component that is common to all the registration speech.

This paper will be organized as follows. Section 2 will discuss the detail of GMM-UBM method for speaker recognition. In section 3, the modification of the GMM-UBM to perform activation keyword detection is explained. Section 4 presents the results of the described scheme tested on SpeechDat-Car database, and section 5 presents the conclusion and future works.

2. GMM-UBM in Speaker Recognition

Here, Gaussian Mixture Model is used to model speaker variability [5]. It consists of N multivariate Gaussian density, with the probability density function.

$$p(y_t | \Theta) = \sum_{i=0}^{N-1} w_i N\{y_t | \mu_i, \Sigma_i\} \quad (1)$$

Θ represents the speaker model, consisting of the weighting parameter w_i , the mean vectors μ_i and the covariance matrices Σ_i . y_t represents the input speech vector, consisting of an array of Mel Frequency Cepstral Coefficient (MFCC) extracted per time frame t . There will be two models that needs to be estimated, the Θ_{UBM} and $\Theta_{speaker}$. The Θ_{UBM} will be estimated using speech data from multiple speakers. It is then adapted using the speech only from the target speaker to obtain $\Theta_{speaker}$ (details of the process will be covered in the

next section). Figure 1 illustrates the adaptation process from UBM model to a target speaker model.

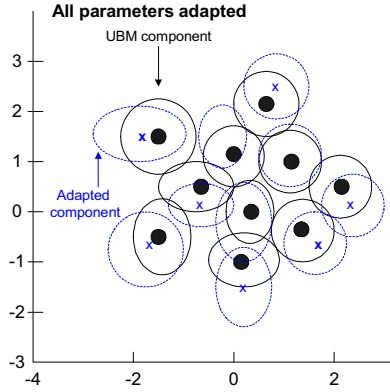


Figure 1: Example of GMM adaptation [2]

During the testing or verification process, the ratio between the log likelihood of the speech being generated by Θ_{speaker} and Θ_{UBM} is calculated, and the speaker is accepted as the target speaker when this result is above certain threshold.

$$LLR = \frac{1}{T} \sum_{t=0}^{T-1} \{ \log p(y_t | \Theta_{\text{speaker}}) - \log p(y_t | \Theta_{\text{UBM}}) \} \quad (2)$$

3. Discussion

The same scheme is now applied for keyword detection purpose. However, this time the Θ_{UBM} is built using all speech from the device user, and Θ_{keyword} is adapted from it using only the utterance of the keyword speech. The process will be elaborated below.

3.1. Training phase

The training phase consists of building the Universal Background Model and the adaptation to reach the word model. The UBM is obtained from the normal speech of the user. This can be easily obtained when he was using his phone. No transcription is necessary as we are only building the acoustic space of the particular user. The speech data are processed per frame, and the MFCC are extracted. These MFCC coefficients are then clustered into the desired number of mixtures. Any clustering methods can be applied for this purpose, for example the K-means clustering method or the hierarchical clustering method. The mean, variance, and weight for each of the clusters are calculated for this UBM. A small number of expectation maximization (EM) can be done to obtain the UBM model.

The next step is to build the word model. This is where the user cooperation is needed to define the keyword and pronounce it 3 times. All the speech is used as training samples to estimate the new GMM for the keyword model. Maximum Likelihood estimation using EM algorithm is used to estimate the model parameters, consisting of weight (w_i), means (m_i) and variance (v_i) (Note that diagonal covariance matrix is used throughout this process, hence the change from Σ_i to v_i). Given the training samples $y = \{y_1, \dots, y_T\}$, where y is the MFCC of the keyword speech.

$$\text{new_}w_i = \frac{\sum_t \alpha_i(t)}{T} \quad (3)$$

$$\text{new_}m_i = \frac{\sum_t \alpha_i(t) y_t}{\sum_t \alpha_i(t)} \quad (4)$$

$$\text{new_}v_i = \frac{\sum_t \alpha_i(t) y_t^2}{\sum_t \alpha_i(t)} - \text{new_}m_i^2 \quad (5)$$

where $\alpha_i(t)$ represents the occupation probability defined as

$$\alpha_i(t) = \frac{w_i N(y_t; m_i, v_i)}{\sum_j w_j N(y_t; m_j, v_j)} \quad (3)$$

The function N here represents the Gaussian likelihood calculation with mean w_i and variance v_i .

Multiple iterations are performed to ensure a dependable keyword model that is separable from the UBM. Besides maximum likelihood, it is also possible to employ maximum a posterior estimation to derive the above keyword model.

Finally, the keyword template is derived from the registration data. As explained earlier, the GMM-UBM modeling would not be able to distinguish between “My TV” and “TV My”. This scheme is devised to tackle that problem. Figure 2 illustrates the keyword template with respect to the GMM for the keyword model. Without this path, there is no difference in terms of LLR score if a person is pronouncing the keyword backwards.

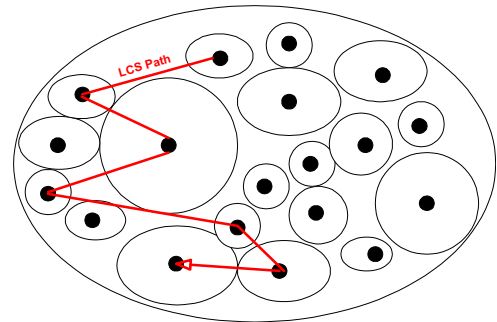


Figure 2: Keyword model GMM with its LCS path.

The template is obtained by performing LCS analysis on the registration keyword. This is a task to find the longest subsequence common to all the registration keywords. Dynamic programming solution is used to obtain the LCS score matrix. The following pseudo code explains the steps in deriving the LCS score in the array *table*, where *len1* and *len2* is the length of the two templates to be compared, and *data1* and *data2* array contains the two keyword template (sequence of index of the best performing mixture from the word model). Basically, the value of *table[i][j]* is extended by 1 if the element of *data1[i-1]* and *data2[j-1]* is equal. Otherwise, the longer value between *table[i-1][j]* and *table[i][j-1]* is retained. The final LCS length can then be found in *table[len1][len2]*. The sequence itself can be derived by backtracking the choices taken when computing *table* score.

```

for (i=0;i<=len1;i++)
for(j=0;j<=len2;j++){
    if ((i==0)||(j==0))
        table[i][j]= 0;
    else if (data1[i-1]==data2[j-1])
        table[i][j]= table[i-1][j-1]+1;
    else
        table[i][j] = ((table[i-1][j] > table[i][j-1])?table[i-1][j]
                        :table[i][j-1]);
}

```

The resulting LCS sequence is again compared with the third template to get the final result. The minimum of the LCS score is kept as threshold. During testing, the LCS result of the test word has to exceed this threshold for it to be accepted as the keyword.

3.2. Testing phase

The testing phase is considerably simpler compared to the training phase. After the test speech is converted to MFCC, the likelihood of the keyword model is calculated using equation 1. Similarly the probability from the UBM is also calculated, and the word is rejected if the log likelihood ratio (LLR) is negative. The LLR calculation is the similar to equation 2, but now what we have is a keyword model (Θ_{keyword}) instead of speaker model (Θ_{speaker}).

If the llr is positive, the sequence of best performing component of the test speech from the word model is passed to the second check. Here LCS analysis is performed the same way as in the testing phase, and the word is accepted as keyword if the value exceeds the threshold defined during training.

3.3. Learning phase

The learning phase is an extra feature to improve the performance of this recognizer. During usage, if a valid keyword is detected, it is used to improve the word model further. The process is similar to the expectation maximization iteration during the training phase, but now it is using the recently acquired speech sample.

4. Experiments

4.1. SpeechDat-Car database

We used a subset of the English SpeechDat-Car database[6] for our experiment. This subset comprises of 75 speakers (25 female and 50 males) recorded in realistic driving condition. The sampling rate is 16kHz, and there are 4 microphone recording simultaneously, positioned at

- A. close talk
- B. on the ceiling of the car near the A-pillar
- C. on the ceiling of the car behind the sun visor
- D. on the ceiling of the car over the mid-console

All the speech from corpus S1-S9 containing phonetically rich sentences is used to build the UBM for each speaker. The keyword is taken from Corpus O7 containing forename and surname. Only the speech from close talk microphone is used

in the training. The front end uses 12 MFCC and the Gaussian model uses 64 mixtures. No noise reduction method is employed in this experiment.

4.2. Results and Discussions

4.2.1. Same speaker test

The testing was performed per speaker on the keywords from each of the 4 microphone, and another 4 random words from the close talk microphone. The first four was to evaluate whether the modeling can handle the differences in recording environment, and the last four was to test whether it can reject a non-keyword. In total there are 600 set of test. Figure 3 shows an example of the results for one speaker. The keyword model likelihood for the first 4 words is higher compared to the last four random words.

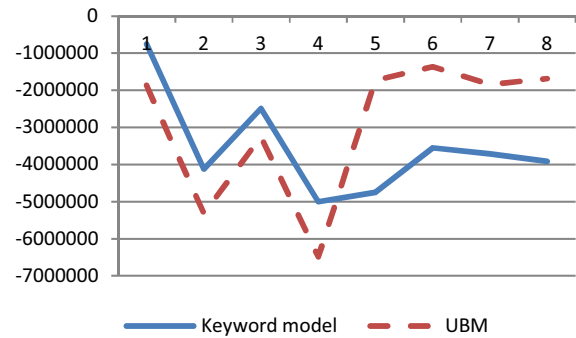


Figure 3: Log likelihood value for the 8 test.

Table 1. Precision/Recall from GMM-UBM method.

	Mic A	Mic B	Mic C	Mic D	Total
Precision	97.40%	98.08%	100%	100%	98.79%
Recall	100.0%	68.00%	84.00%	74.67%	81.67%

Table 1 summarizes the results before the LCS check. The scheme reaches recall rate of 81.67%, with most of the misses coming from the distance microphone B and D. In terms of precision however, it is close to perfect. Note that these results still have to go through LCS check and in this experiment, all the false positive outside the 300 true cases was rejected at that point. However, the LCS check also has an impact on the false negative. As shown in table 2, although the precision rate hits 100% after the LCS check, the recall rate drops to 71%, again mainly the error comes from the distance microphone B and D. This could be due to the fact that in this experiment, no noise reduction method and no channel compensation is performed.

Table 2. Precision/Recall after LCS.

	Mic A	Mic B	Mic C	Mic D	Total
Precision	100.0%	100.0%	100.0%	100.0%	100.0%
Recall	100.0%	53.33%	72.00%	58.67%	71.00%

Figure 4 shows an example of an LCS score for 5 different speakers. The SpeechDat-Car database only gives one utterance of the keyword (the names from Corpus O7), therefore we only use that single registration path as our template. The first three results in figure 3 are the LCS score

for the keyword from the 3 distance microphone. They are much higher than the last 4 results which come from random words. This diagram is just for illustration only, since most of those random words are already rejected due to the negative LLR and never made it to LCS check.

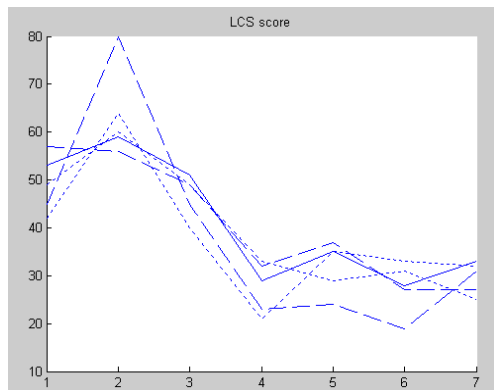


Figure 4: LCS score for 5 different speaker

4.2.2. Different speaker test

Our second experiment deals with the case where a foreign user is trying to use the system. Similar 8 sets of tests are used, but this time they do not come from the valid user. The first four is an invalid user pronouncing the keyword from each of the microphone, and the last four is random words. These 600 tests should all be rejected since they do not come from a valid user.

Table 2. False accept of Invalid User

	Before LCS	After LCS
Keyword	19.67%	2.83%
Random words	3.50%	1.33%
TOTAL	23.17%	4.17%

Table 2 shows the false accept rate before and after LCS test. When an unauthorized user is pronouncing the keyword, there is 19.67% false acceptance from the LLR test, but this error is reduced to 2.83% by the LCS test. The false accept for random words from an invalid user are much lower and only amount to 1.33% after both LLR and LCS test. The total false accept rate for this system is 4.17%

5. Conclusions

We presented a new method for activation keyword detector that deviates from the common speech recognition technique such as HMM and DTW. We adapted the GMM-UBM scheme from the field of speaker recognition to perform this task. The target application is a personal device, for example a phone, that would be activated upon the recognition of the keyword being pronounced by its valid user. The fact that the device would have full access to the user speech is advantageous in this scheme as the UBM would need to be built from the voice of the user. The keyword model is then adapted from the UBM during the registration phase when the user gives sample of the keyword. To solve the problem that this modeling does not contain a temporal axis, we introduce a second layer of check using LCS algorithm by keeping the path and LCS score of the registration keyword.

We tested our scheme using a subset of SpeechDat-Car database. The corpus containing phonetically rich sentences

are used to build the UBM, and the keyword is a unique name pronounced by each user. Our proposed method gives precision rate nearing 100%. This good rejection rate is due to the two levels check introduced. The recall rate reached in this experiment is 71%. Although the GMM is known to have robustness features in the speaker recognition field, the absence of noise reduction in our experiment could have caused this degradation. The system is also tested against invalid users who try to use the system by pronouncing the keyword. The effectiveness of LCS test is evident here where it manages to reduce the false accept rate from 19.67% to 2.83%. The total false accept rate for an invalid user in this system is 4.17%.

Another advantage of this scheme is its learning capability as the system is used. The keyword detected from the valid user can be continuously used to improve the keyword model. Unlike DTW method which learns by keeping all the keyword templates and occupying more memory space, our scheme would simply adapt the keyword model without consuming any extra storage.

The introduction of a more noise robust features for this modeling with some channel compensation scheme will be part of our future work.

6. References

- [1] S. Furui, "Selected topic from 40 years of research in speech and speaker recognition," *INTERSPEECH 2009*, Brighton, UK, pp. 1-8, 2009.
- [2] Tomi Kinnunen, Haizhou Li, "An Overview of Text-Independent Speaker Recognition: from Features to Supervectors," *Journal of Speech Communication*, vol. 52, Issue 1, January 2010.
- [3] Chang Huai You, Kong Aik Lee, Haizhou Li, "GMM-SVM Kernel With Bhattacharyya-Based Distance for Speaker Recognition," *IEEE Transactions on Audio Speech and Language Processing*, 2010.
- [4] National Institute of Standards and Technology, Speaker Recognition Evaluation, <http://www.itl.nist.gov/iad/mig/tests/spk/>
- [5] D.A. Reynolds, T.F. Quatieri, R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19-41, 2000.
- [6] Asuncion Moreno, Borge Lindberg, Christoph Draxler, Gael Richard, Khalid Choukri, Stephan Euler, Jeffrey Allen, "SPEECHDAT_CAR. A Large Speech Database For Automotive Environments," *International Conference on Language Resource & Evaluation*, Athens, 2000.