**Piush Karmacharya**

**Department of Electrical and Computer Engineering, Temple University**

**A short Literature Review on**

**Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams**

***tub86339@temple.edu***

1. **Summary**

This report is a review of paper *Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams by Yaodong Zhang and James R. Glass*. As the name suggests, an unsupervised method of Keyword Spotting (KWS) is proposed that is based on Dynamic Time Warping (DTW) of Gaussian Posteriorgrams unlike traditional HMM based systems. Without any transcription information, a Gaussian Mixture model is trained on a speech database. For given keyword or keywords, posterior probability is computed and segmental DTW is used to identify presence of the keyword in the utterance. Detection decision is based on ranking the distortion scores for all segments. The system is first tested on TIMIT corpus and later evaluated on the MIT lecture corpus. The rest of the report is organized as follows: Section II contains brief introduction to keyword spotting problem with supervised/unsupervised learning viewpoint. Section III contains brief description of the terminologies used in the research. Section IV describes some other ongoing relevant researches followed by experimental setup and results in section V. Lastly, the report is concluded with discussion over the methodologies and personal opinion.

1. **Introduction**

Keyword spotting (KWS) is a branch of speech processing that deals with identifying certain keywords in a long utterance. In present context when people are trying to make human-machine interaction seamlessly natural, the scope of this field is even bigger because human conversation contains not only irrelevant words but also non-intentional sounds like cough, exclamations and noise. If we can extract only the embedded information, computation can be much efficient and robust. Some examples of these kinds of systems can be voice dialing, voice command systems, audio search engines etc. Keyword spotting also has many direct applications such as audio document retrieval and covert speech surveillance systems etc.

A Keyword Spotting System (KWS) is given a keyword or multiple instance of the keyword and a stream of utterance which is basically the search space where it has to locate the keyword if present. One way to identify keyword in the utterance is to parse the utterance through a Large Vocabulary Continuous Speech Recognition (LVSCR) System, which are usually Hidden Markov Model (HMM) based systems, and get transcription for the utterance. The task is then reduced to text recognition which is rather simpler as compared to the original problem of spoken keyword. But this requires training the system on an infinitely large dictionary of words and huge language model which is rather impractical for small scale systems. An alternative approach is to consider the speech utterance to be composed to keywords and non-keywords and using garbage or filler models to characterize non-keywords. Some systems add extra models for representing non-speech events such as silence or coughing. Both keyword model and garbage model are built from concatenation of phoneme sub-models. For these systems, training data does not require to be transcribed in word level but has to be correctly labeled in phone level. Although adaptive systems are being designed that can later be trained on un-transcribed data, the initial design phase of the system requires considerably large amount of transcribed data. This creates a big problem in cross-language or language independent system design. Although development of annotated linguistic resources is progressing substantially, the result fall dramatically short of covering nearly 7,000 human languages spoken around the globe. The annotation work is not only time consuming, but it also requires linguistic expertise for providing necessary annotations which can be barrier to new languages.

This research is focused on spoken term detection focused on above-mentioned environment where standard techniques for keyword detection prove infeasible. As we enter an era where digital media is created and accumulated at a rate that far exceeds our ability to annotate it, it is natural to question how much can be learned from the data alone, without any kind of supervised input. This paper presents a completely unsupervised learning framework to address the problem of spoken term detection. For that a Gaussian Mixture Model (GMM) is trained to represent each speech frame with a Gaussian Posteriorgram. Given one or more instance of keyword to be detected in the utterance, Gaussian posteriorgrams between keyword(s) and test utterance is compared using Segmental Dynamic Time Warping (SDTW). Decision is made based on the distortion score on the test utterances and best matching candidate are selected as keyword hits.

1. **System Design**

**Development of Acoustic Model**

For acoustic modeling of the speech data Gaussian Mixture Models (GMMs) are developed for each speech frame for entire training data. For this Mel Frequency Cepstral Coefficients (MFCCs) for each frame is computed. MFCCs are most commonly used spectral feature set used in speech recognition derived as follows:

1. Speech signal is divided into short frames over the range it can be considered stationary.
2. Fourier transform of each frame of the signal is obtained.
3. Mel scaling is applied to the Fourier coefficients using triangular overlapping windows. These filters are linearly spaced for lower frequency while log-spaced filters at higher frequency.
4. Logs of the powers are computed at each of the mel frequency.
5. Discrete cosine transform (DCT) is computed of the mel log power. These mel-scaled scaled cepstrum exist in a domain referred as *quefrency* which as the same unit as time.
6. The MFCCs are the amplitudes of the resulting spectrum.

This way we end up with multidimensional MFCC feature vector for each frame. Consider the first dimension of these vectors for entire keyword ‘ONE’ as shown below.



Figure 1. Distribution of the first dimension of MFCC feature vectors for digit ‘ONE’

Figure 1 shows the distribution of the first dimension of MFCC feature vectors extracted from the training data for the digit ‘one’. We could use a distribution to fit the PDF, but the distribution looks quite arbitrary, and standard distributions do not provide a good fit. One solution is to fit a Gaussian mixture model (GMM), a sum of weighted Gaussians as shown in Figure 2.

The complete Gaussian mixture density is parameterized by the mixture weights, mean vectors, and covariance matrices from all component densities. GMM parameters are estimated for entire unlabelled training data initialized by K-means algorithm.



Figure 2. Approximation of feature vectors using weighted sum of Gaussians

For a given keyword, MFCC feature vector is found in similar manner. The objective is to find a Gaussian posteriorgram distribution for our GMM. Given GMM models and a test feature vector, the log-likelihood value of that feature set belonging to a particular GMM is given by the posterior probability. Posterior features have been widely used in template-based systems. Gaussian posteriorgram is a probability vector represtnting the posterior probabilities of a set of Gaussian components for a speech frame.



Figure 3. An example of phonetic posteriorgram representation for the spoken phrase ‘basketball and baseball’

A sample of phonetic posteriorgram is shown in Figure 3 for illustration where each element of the vertical strip represents posterior probability for each frame belonging to that particular phone. Method proposed in this paper however uses pure Gaussian Posterirorgram and not phone model.

Formally, if we denote a speech utterance with n frames as S = (s1,s2,…,sn), then the Gaussian posteriorgram (GP) is defined as:

GP(S) = (q1,q2,…qn)

Each qi vector can be calculated by

qi = ( P(C1|Si), P(C2|si), …. , P(Cm|si) )

where Ci represents the i-th Gaussian component of a GMM and m denotes the number of Gaussian components.

**Template Matching**

Dynamic time warping is used to find a non-linear alignment path between two sequences that optimizes the distance between the two sequences. It is obtained by time scaling one of the signals non-linearly so that it aligns close to the other. It is extremely efficient time-series similarity measure. It minimizes the effects of shifting and distortion in signals allowing elastic transformation in time scale. It is one of the earliest approaches to isolated word recognition. For keyword spotting, prototype of a keyword is stored as a template and compared to each word in the incoming speech utterance.

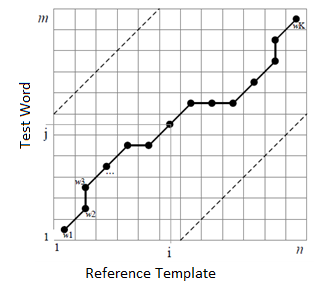


Figure 4. Dynamic Time Warping

As shown in figure 4, feature vector for reference keyword and input keyword are arranged along two side of the grid. In this case, the reference word of length n is arranged along the horizontal axis and test word of length m along the vertical axis. Each block in the grid is the distance between corresponding feature vectors. The best match between these two sequences can be computed from the path through the grid which minimized the total cumulative distance between them as shown by the dark line in Figure 4 Total distance between the test and the reference data is the sum of distance along the path.

From the figure above, it is apparent that the number of possible paths through the grid grows exponentially with the length of the word. Applying some, we can limit the possible paths making the computation feasible. These constraints basically limit the possibility of infinite path making computation more efficient. These constraints are:

• Monotonic condition: The path has to be monotonically increasing function and cannot turn back. If i and j are the indices denoting reference and test features, then both i and j either stay the same or increase.

• Continuity condition: There can be no break in the path. i and j can only increase by 1 on each step.

• Adjustment window condition: The optimal path should not wander far away from the diagonal. This constraint is based on the fact that the ideal path is along the diagonal. The maximum deviation is called the window length.

• Boundary condition: The path should start at the bottom left and end at top right. This is logical if we are comparing two complete words.

• Slope constraint condition: The path should neither be too steep not too shallow. Too steep of shallow gradient causes an unrealistic correspondence between a very short pattern and a relatively long pattern.

There are different variations to these constraints depending on the application of the algorithm. For connected word recognition where it is hard to precisely indicate the word boundary, afore mentioned boundary condition is changed to cover some range of possible beginning and end of a word. Similarly in some cases maximum allowed deviation is selected over the most recent optimal path instead of the diagonal to increase the efficiency.

This research in particular is about keyword spotting in continuous speech utterance. So the boundary of each keyword is unknown. Therefore, a modified version of DTW called Segmental-DTW (SDTW) is proposed. SDTW takes as input tow continuous speech utterances and finds matching pairs of subsequences. Given two utterances, X and Y, we can represent each as a time series os spectral vectors, (x1, x2, …, xN) and (y1, y2, …, YN), respectively. The optimal alignment path between X and Y, Ф, can be computed, and accumulated distortion between the two utterances along that path dФ(X, Y), can be used as a basis for comparison. The warping relation can be written as sequence of ordered pairs

Ф=( ik, jk) k = 1, … T

that represents the mapping xi1 ↔ yj1, xi2 ↔ yj2, …, xiN ↔ yjN

The proposed solution is a segmental variant of DTW that attempts to find subsequences of two utterances that align well to each other. It compromises of two main components: a local alignment procedure which produces multiple warp paths that have limited temporal variation, and a path trimming procedure which retains only the lower distortion regions of an alignment path. The major modification over traditional DTW for this implementation is the imposition of two constraints. The first one is the adjustment window constraint to restrict the shapes that a warping path can take as already defined in the previous section in the report. Second, multiple alignment paths are allowed for the same two input sequences by employing different starting and ending points in traditional DTW search algorithm.

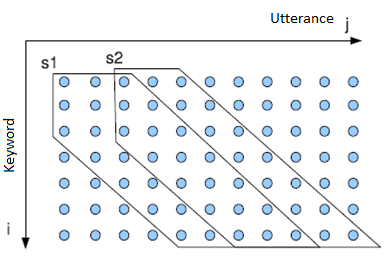


Figure 5 Segmental DTW for adjustment window size 2

As shown in figure 5, the adjustment window condition not only restricts the shape of the warping path but also the ending coordinate for a given starting point and thus divides the search grid into finite segments for generating multiple alignment paths. For example, if the starting coordinate (i1, j1) is (1,1) and the end iend = m, then jend ϵ (1+m-R, 1+m+R). As a result, the difference matrix can be divided into several continuous diagonal segments of width 2R+1. In order to avoid redundant computation and taking into account warping paths across segmentation boundaries, use of overlapping sliding window seems plausible as indicated by starting points s1 and s2 in Figure 5.

1. **Related Work**

There has been number of research towards unsupervised training of keyword spotting systems. Some of them involve modeling subword-unit, ergodic HMMs that could be trained on unsupervised data or phonetic posteriorgrams. Most of the methods are based on breaking word into smaller units like phones or letters and mapping a keyword into connection of those units. The major advantage of those systems is that we can use non-keyword data to model any keyword making it applicable in out-of-vocabulary words too. Some method involve using unsupervised data to train ergodic HMM model and use one or more instance of keyword and convert each instance of ergodic HMM into a series of HMM states. The main difference of Ergodic HMM (EHMM) from traditional left-to-right HMM is that all states of the EMMs are mutually connected. The HMM state sequences are then combined to create a conventional HMM to represent the keyword.

Methods based on modified DTW have been investigated not only for keyword spotting but also for Spoken document classification and other kind of unsupervised pattern discovery. This is particularly important in cases where transcribed data are not available like because transcribing data costs lots of time and money. Some of the approaches are phone or syllable based while others are frame based. For phone based approaches, some prior knowledge is required in the form of phonetic posteriorgrams. A phonetic posteriorgram is defined by a probability vector representing the posterior probabilities of a set of pre-defined phonetic classes for a speech frame. This requires running an independently trained phonetic recognizer in the data. Test keyword is first converted into a series of phonetic prosterigram for each frame. Then modified DTW is used in order to identify similar pattern in given utterance using distortion scores. For improving the performance, multiple instance of keyword is used with user-driven feedback system.

Other modifications to classical DTW algorithms have also been proposed for keyword spotting. The Unbound Dynamic Time Warping (UDTW) algorithm method uses possible alignment points called the synchronization points between two segments where time warped matches are searched for. This eliminates exhaustive computation of all the cost matrix start. Then a forward-backward dynamic programming algorithm is used to find exact start-end alignment points. Two additional constraints are implied to DTW. First a minimum length is defined. Secondly a maximum time warping of 2X and minimum of X/2 is applied by defining proper local constraints. The main challenge to this approach is finding the synchronizing points. Given that a matching segments can occur anywhere within the two sequences compared, exploiting the minimum length constraint, we take fewer locations of SP (synchronization points) while ensuring that if there is a matching segment, it will be found. Both the accuracy and the speed of the algorithm depend on selection of SP; sparse SPs increases the processing speed at the expense of possibly missing matching segments, whereas dense SP’s are computationally more expensive to process.

1. **Experimental Setup**

The proposed approach is similar to phonetic posteriorgram described earlier. But in this case direct GMM is computed over unsupervised training data instead of using a phonetic recognizer. As a result, the posteriorgram so obtained becomes a Gaussian posteriorgram instead of phonetic posteriorgram. Then SDTW is applied to compare Gaussian posteriorgrams between keyword and test utterance to identify possible match. First a GMM is trained over unsupervised training data to produce a raw posteriorgram vector for each speech frame. Since the training is unsupervised, induction of noise in the training data can easily generate an unbalanced GMM because of their large variance. This results in posteriorgram that cannot discriminate well between phonetic units. In order to resolve this issue, a speech detector was used to extract only speech segment and train the GMM on speech segments only. To avoid approximation errors, a probability floor threshold is applied setting posterior probabilities less than Pmin to 0. The vector is then renormalized to set summation of each dimension to 1. This would create many zeros in the posteriorgram. To avoid that, a discounting based smoothing strategy is applied to move a small portion of probability mass from non-zero dimension to zero dimensions. For each Gaussian posteriorgram vector q, each zero dimension *zi* is assigned to *zi = λ/ Count (z)* where *Count(z)* denotes the number of zero dimensions. Each non-zero dimension vi is changed to *vi = (1- λ)vi*.

In order to apply SDTW, the difference between two Gaussian posterior vectors *p* and *q* is defined as

D(*p*,*q*) = - log (*p* . *q*)

Since both p and q are probability vectors, the dot product gives the probability of these two vectors drawing form the same underlying distribution. Applying Segmental DTW over the test utterance results (n-1)/R warping paths where n is the number of frames in the utterance and R is the window size. Each path has its corresponding distortion score for each test utterance. Ones with minimum distortion are selected as candidate region of keyword occurrences. In case where more than one instance of a keyword is available, a scoring strategy is applied to calculate the final score for each test utterance taking into account the contribution of all keyword. Given multiple keyword samples and a test utterance, a reliable warping region is the one with most of the minimum distortion warping paths of the keyword samples aligned. So a region with smaller number of alignments to keyword samples is considered less reliable.

An efficient binary tree is used to count the number of overlapped alignment regions and the ones with only one sample alignment are discarded. With rest of the alignments, a fusion method is applied. Formally, if si samples of keyword are aligned to a region rj in the utterance, the final distortion score is:

Varying α between 0 and 1 changes the averaging function from a geometric mean to an arithmetic mean.

For deriving the GMM models, speech utterance was divided into 25ms frames with a 10ms window shifting. 13 order MFCC coefficients were extracted for each segment. All the MFCC frames were used to train a GMM with 50 components which was used to decode both the training and test frames to produce Gaussian posteriorgram representation.

1. **Experimental Results**

System design and first phase evaluation was carried on TIMIT corpus which consists of read speech in quiet environment. After achieving satisfactory performance on this corpus, second stage system testing was performed on MIT lecture corpus which consisted of more than 300 hours of speech data recorded in general seminar and classroom environment using a lapel microphone. Because of these environmental factors, there were many non-speech artifacts including background noise, laughter etc. To avoid unbalanced GMMs because of the noise, an independently trained speech detection module was used to filter out non-speech segments. The speech detection module used does not need prior information of data or any kind of language module, allowing the system to still be unsupervised.

Three evaluation metrics were used to measure the performance; P@10: the average precision for the top 10 hits, P@N: the average precision for top N hits, where N is equal to the total number of occurrences of each keyword in the test utterance, and EER: the average equal error rate at which the false acceptance rate is equal to false rejection rate in the ROC (Receiver Operator Characteristic) curve. Basically three parameters: the smoothing factor *λ*, the window size of the SDTW *w* and the weighting factor *α;* are varied to study the performance variation and find optimal parameters for the system.

For TIMIT corpus, experiment was conducted on the standard training set of 462 speaker and 3696 utterances and testing set was composed of aa8 speaker and 944 utterances. Total size of vocabulary was 5851 words. 10 words were randomly selected with variety of numbers of syllables for testing purpose. Below are the plots of results obtained.



Figure 6. Result for varying smoothing factor

Smoothing factor *λ* ranges from 0.1 to 0.00001 and so is plotted in log scale keeping window size *w*=6 and scoring factor α=0.5. It can be seen that the best setting in terms of EER and P@N is achieved as *λ =* 0.0001.

Similarly window size for SDTW *w* was varied keeping *λ=*0.0001 and α =0.5. Results show optimal result was obtained for *w*=6 as seen in Figure 7. It is in accordance with our assumption that small DTW window size restricts the warp match between keyword references and the test utterance lowering the performance. Allowing larger window size allows warping paths with excessive time difference, which would also affect the performance.



Figure 7. Result for varying SDTW window size

Similar experiment on weighting factor gave optimal performance at α =0.5. Another important parameter that could largely affect the overall performance of the system is the number of Gaussian components in the GMM. Keeping other parameters at their optimal values obtained from above experiments, number of Gaussian components was changed.



Figure 8. Result for varying number of Gaussian components

Result in Figure 8 suggests that the number of Gaussian components in the GMM training has major contribution on the performance. When the number is small, GMM training may suffer from under fitting problem causing low detection rate. Using too many Gaussian components would make the system more sensitive to variations in the training data which could result in generalization errors on the test data. Optimal number of components would be an approximation of number of underlying broad phone class in the language. Based on the results, 50 Gaussian components were chosen for optimal performance.

Experiment on MIT lecture corpus was performed in similar manner. The training data set contained 57,351 utterances and test set contained 7,375 utterances with total 27,431 words on each set. 30 keywords were selected in random for evaluation, all of them occurring more than 10 times in both training and test sets. System parameters were set to optimal values obtained from previous experiments conducted on TIMIT database. Given below are the results of the experiment.

|  |  |  |  |
| --- | --- | --- | --- |
| # Samples | P@10 | P@N | EER |
| 1 | 27.0% | 17.3% | 27.0% |
| 5 | 61.3% | 33.0% | 16.8% |
| 10 | 68.3% | 39.3% | 15.8% |

Table 1. Performance for different number of sample keywords

As it can be seen in Table 1, there is dramatic improvement on the performance when more than one instance of test keyword is fed to the system. This is mainly due to voting based merging strategy. Improvement is slower for over more than five instances. An interesting result derived from the experiment was that system performance for words with more syllables tends to be better than ones with fewer syllables. This result suggests that keywords can be categorized as good keywords and bad keywords according to their discriminability for a given keyword spotting system.

1. **Discussion:**

A template based Keyword Spotting system has been proposed in this paper that requires no supervised learning. Speech transcription requires lot of time and money. Moreover, considering the diversity of languages spoken over the world, collecting enough data to train state of art statistical systems and transcribing them is not feasible. Considering these scenarios, unsupervised learning is an interesting research filed in speech recognition. Although various adaptive statistical systems have been developed, they still require considerable amount of well transcribed data for initial training.

The system is able to learn parameters from available data and identify keyword in a given speech utterance provided one or more samples of test keyword. The idea is to identify feature patterns in the given utterance that is common to the keyword. Dynamic Time Warping is modified in a way that it does not require precise word boundary. Speech utterance is divided into segments each with its warping path and a distortion score. This causes the search space to be much larger requiring more computation. But since the computation is straightforward it should still be practically feasible. But no comparison is made based on computation efficiency or time as to test system performance on real-time data. Identification is based on distortion score for each segment on the utterance; one with lowest distortion is the candidate keyword. If more than one instance of test keyword is available, weighted score is computed maximizing the system performance.

A Gaussian Mixture Model is trained in unsupervised manner and each frame in the test keyword is represented in terms of Gaussian posteriorgram. The GMM is initialized by K-means algorithm and since the clustering is random, system performance can vary slightly each time it is trained even on same training data. For this reason each experiment is ran five times and best result is presented. But there is no specification about the number of clusters or details of the clustering process in the paper and no suggestion is made on keeping the system consistent each time the system is trained.

Once posterior probability is calculated for the utterance, a probability floor threshold is applied to eliminate dimensions with posterior probability less than the threshold setting it to zero. This requires the entire vector to be normalized to set total sum to one. Again a discounting based smoothing strategy is applied possibly to convert the zero dimensions to non-zero value. Even though there is description of what is being done on the poster probabilities, no convincing description is found that explains why this is being done and what its impact on overall system is.

Despite choosing optimal parameters derived from multiple experiments, the overall system performance is still pretty low as compared to state-of-art systems. But since this approach is in early development state, there is still more room for further improvement. The authors have proposed an adaptive learning technique for optimal number of Gaussian components which controls the discrimination capability and has direct impact on system performance. The proposed system cannot handle Out Of Vocabulary (OOV) words which is an area of future enhancement. System performance on other languages is still to be studied in order to prove language independency of the system.

**References**

1. Yaodong Zhang, and J. R. Glass. 2009. Unsupervised spoken keyword spotting via segmental DTW on gaussian posteriorgrams. Paper presented at Automatic Speech Recognition & Understanding, 2009. ASRU 2009.
2. Anguera, X., R. Macrae, and N. Oliver. 2010. Partial sequence matching using an unbounded dynamic time warping algorithm. Paper presented at Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference
3. Hazen, T. J., W. Shen, and C. White. 2009. Query-by-example spoken term detection using phonetic posteriorgram templates. Paper presented at Automatic Speech Recognition & Understanding, 2009. ASRU 2009.
4. Myers, C., L. Rabiner, and A. Rosenberg. 1980. An investigation of the use of dynamic time warping for word spotting and connected speech recognition. Paper presented at Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80. .
5. Park, A. S., and J. R. Glass. 2008. Unsupervised pattern discovery in speech. *Audio, Speech, and Language Processing, IEEE Transactions on* 16 (1): 186-97.
6. Peng Li, Jiaen Liang, and Bo Xu. 2007. A novel instance matching based unsupervised keyword spotting system. Paper presented at Innovative Computing, Information and Control, 2007. ICICIC '07. Second International Conference
7. Sakoe, H., and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on* 26 (1): 43-9.
8. Yaodong Zhang, and J. R. Glass. 2010. Towards multi-speaker unsupervised speech pattern discovery. Paper presented at Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference.
9. http://www.mathworks.com/company/newsletters/digest/2010/jan/word-recognition-system-matlab.html